NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

JOINT MOBILE NETWORK OPERATIONS: ROUTING DESIGN AND QUALITY OF SERVICE CONFIGURATION

by

David K. Norton

September 2007

Thesis Advisor:	Geoffrey Xie
Second Reader:	John Gibson

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 2007	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Joint Mobile Network Operations: Routing Design and Quality of Service Configuration	5. FUNDING NUMBERS
6. AUTHOR(S) David K. Norton	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING/MONITORING AGENCY REPORT NUMBER N/A

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Current inter-Service military networking is inefficient and lacks the desired level of Joint interoperability. Generally, the different Service branches build stove-piped networks that do not allow sharing of resources with the other branches. This approach is taken because the individual networks do not see the benefits of interconnectivity as worth the effort required to build secure, stable, and operationally effective network solutions.

The Joint Mobile Network Operations (JMNO) project seeks standard solutions to the networking challenges of tactical military units. Through the publication of these standards, the intent is to reduce the complexity of finding networking solutions. This, in turn, reduces the perceived cost of inter-Service networking, making it more attractive to military units.

This thesis provides some specific solutions that can be included in the JMNO standards. It examines network routing and provides recommendations for protocol selection and configuration. It also recommends implementing certain Quality of Service (QoS) controls to make more efficient use of available bandwidth, to provide preferred handling of critical time-sensitive traffic, and to provide individual networks a means of protecting their links from misuse by mobile units.

14. SUBJECT TERMS Network, Routing, Mobile, Border Gateway Protocol (BGP), Dynamic Host Configuration Protocol (DHCP), Quality of Service (QoS), Differentiated Services (DiffServ)	15. NUMBER OF PAGES 153
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

JOINT MOBILE NETWORK OPERATIONS: ROUTING DESIGN AND QUALITY OF SERVICE CONFIGURATION

David K. Norton
Captain, United States Marine Corps
B.S. (Political Science), Oregon State University, 1998

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2007

Author: David K. Norton

Approved by: Geoffrey Xie
Thesis Advisor

John Gibson
Second Reader

Peter J. Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Current inter-Service military networking is inefficient and lacks the desired level of Joint interoperability. Generally, the different Service branches build stove-piped networks that do not allow sharing of resources with the other branches. This approach is taken because the individual networks do not see the benefits of interconnectivity as worth the effort required to build secure, stable, and operationally effective network solutions.

The Joint Mobile Network Operations (JMNO) project seeks standard solutions to the networking challenges of tactical military units. Through the publication of these standards, the intent is to reduce the complexity of finding networking solutions. This, in turn, reduces the perceived cost of inter-Service networking, making it more attractive to military units.

This thesis provides some specific solutions that can be included in the JMNO standards. It examines network routing and provides recommendations for protocol selection and configuration. It also recommends implementing certain Quality of Service (QoS) controls to make more efficient use of available bandwidth, to provide preferred handling of critical time-sensitive traffic, and to provide individual networks a means of protecting their links from misuse by mobile units.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AF	Assured Forwarding
AFFOR	Air Force Forces
ARFOR	Army Forces
ARG	Amphibious Ready Group
AS	Autonomous System
BGP	Border Gateway Protocol
C2	Command and Control
CRC	U. S. Air Force Control and Reporting Center
CVBG	Carrier Battle Group
DF	Default Forwarding
DHCP	Dynamic Host Configuration Protocol
DiffServ	Differentiated Services
DISN	Defense Information System Network
DMS	Defense Message System
DRSN	Defense Red Switch Network
DSCP	Differentiated Services Code Point
DSN	Defense Switched Network
DTH	DMS Transition Hub
DVS-G	Defense Video Services – Global
eBGP	External Border Gateway Protocol
EF	Expedited Forwarding
EIGRP	Enhanced Interior Gateway Routing Protocol

FIFO	First-In, First-Out
iBGP	Internal Border Gateway Protocol
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IOS	Internetwork Operating System
IP	Internet Protocol
IRC	Internet Relay Chat
JMNO	Joint Mobile Network Operations
JTF	Joint Task Force
JWICS	Joint Worldwide Intelligence Communications System
Kbps	Kilobits per second
KBps	Kilobytes per second
MAC	Media Access Control
MARFOR	Marine Forces
MTU	Maximum Transmission Unit
NAVFOR	Navy Forces
NIPRNET	Unclassified but Sensitive Internet Protocol Router Network
OSPF	Open Shortest Path First
PHB	Per Hop Behavior
POP3	Post Office Protocol 3
QoS	Quality of Service
RFC	Request For Comments
RIB	Routing Information Base
RIP	Routing Information Protocol

RSVP	Resource ReSerVation Protocol
SIPRNET	Secret Internet Protocol Router Network
SMTP	Simple Mail Transfer Protocol
SNMP	Simple Network Management Protocol
STEP	Standardized Tactical Entry Point
TACC	Marine Corps Tactical Air Command Center
TCP	Transmission Control Protocol
ToS	Type of Service
TTP	Tactics, Techniques, and Procedures
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
VTC	Video Teleconferencing
WFQ	Weighted Fair Queuing

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

My deepest, heartfelt thanks go to my wife, Jenny, and our children: Crystal, Brian, Alex, and Andrew. They provided me the will and encouragement to continue working when things seemed tough. More specifically (and entirely appreciated) is the fact that they conducted an entire duty station move without me. This allowed me to spend a summer focusing on the research and writing of this thesis. They continue to give me a reason to strive for success.

Professor Geoff Xie provided enormous amounts of prodding and guidance that allowed me to get started and kept me moving toward the goal. I particularly appreciate the way he was able to help me find solutions when I was entirely frustrated with specific roadblocks. His easy-going nature and common-sense approach to problem solving helped make the research process much more bearable.

Professor John Gibson assisted in numerous ways along the path. He provided my first practical experiences in building network solutions for this thesis. Along the way he was always available as a sounding board for ideas and never failed to come up with an approach I had not considered previously. He always had a cup of coffee available and was willing to discuss any topic, even when I just needed a break from the research.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Operating under the traditional military communications structure, the different Service branches generally build stove-piped networks that do not allow sharing of resources with the other branches. If an Army unit moves independently across the battlefield, it needs to bring the long-haul communications equipment to reach back to its parent Service. Even if there is a nearby Air Force or Marine unit, the mobile unit cannot tie into their existing network for connectivity. This adds to the logistic footprint and support required for individual units, thereby reducing their mobility. It also creates more competition for limited spectrum and satellite resources as separate communications links are built for each unit. The problem, consequently, is a question of how to share network resources without causing damage to their integrity, security, or operational effectiveness.

A. JOINT MOBILE NETWORK OPERATIONS OVERVIEW

The Joint Mobile Network Operations (JMNO) working group has been formed to find solutions to improve usage of network resources between the various Service branches.

> Joint Mobile Network Operations (JMNO) is a solution to overcome the protocol challenges of mobile networking in joint maneuver warfare. It improves the ability of joint force tactical units (brigade and below or equivalent) to communicate directly with each other across Service lines and to access information resources and network services when crossing Service network boundaries. JMNO will enable such tactical units to act as endpoints for lateral links between the services, enabling specific IP traffic to traverse such links to reduce latency between C2 systems, and also reduce IP traffic going through higher headquarters [1].

Among the many areas it is investigating is the ability to have mobile users utilize the network resources of another Service when the user cannot connect directly to its home Service. An example could be a Marine unit transiting through an

area under Army control. Using current tactics, techniques, and procedures (TTP), the Marine unit would bring organic or augmented communications equipment to tie back to its headquarters. This adds to its logistics footprint and reduces the speed and mobility of movement. It also creates problems with competition for scarce spectrum and/or satellite resources due to the increased transmission path requirements. Rather than having to build an isolated link to the Marine headquarters, it should be able to use the existing Army network to provide the communications required. However, there are many problems with actually implementing this concept. Security and policy issues are extremely critical and are being addressed separately from this thesis.

Another issue is the host Service's concern regarding the impact of the mobile user on its own network. In the example, the Army is unlikely to allow the Marine unit to connect without some assurance that its network will not be adversely affected. There is a distinct need, therefore, for some type of control measure that allows the host Service to maintain a performance level on its own network. There is another important consideration on the other side. The mobile unit will want to ensure that it receives enough access to network resources to accomplish its mission. In particular, there may be time-sensitive traffic that should receive special handling to ensure it is operationally effective. A conflict would arise when the requirements for the host network and the mobile user combine to be greater than the capacity of the network. In this case, some type of negotiation mechanism would need to exist that resolves the issue.

In order to provide any standard networking solution, it is important to understand the framework of the environment on which it is to be implemented. JMNO has developed some basic operating structures that provide a starting point for a routing solution. This framework that JMNO has defined affects the design and flexibility of the solutions that might be implemented. The equipment used, the current communications infrastructure, and the configuration of network enclaves are affected by these policy decisions. The following Sections describe portions of the operational framework defined by JMNO.

1. Routing Equipment

Currently all tactical routers used by the military are manufactured by Cisco [1]. The JMNO solution, for that reason, seeks to capitalize on this commonality by finding standard practices that can be used on Cisco equipment. All configurations and protocols used in this thesis are implemented on Cisco routers. There remains a possibility, however, that in the future the router sources might not be entirely homogeneous. To allow the solutions to remain as flexible as possible, industry standard protocols are generally preferred over proprietary ones. Where proprietary methods are employed, they are intended to be isolated so that a later change would not alter the overall solution.

2. JMNO Network Tiers

The draft JMNO standards document describes the current military networking structure [1]. It divides the operational network into tiers in order to allow solutions to be aimed at the appropriate level(s) of structure. These tiers follow a hierarchical structure that reaches from the lowest levels of military elements up to high level communications networks. Although the tiers vary in description between communications infrastructure components, application domains, and organizational units, the structure defined by JMNO provides a standard reference within its problem area for differentiating between the operational areas of a military network. The interoperability of these tiers is seen not in the structure of the various layers, but in their shared use of Internet Protocol (IP). Table 1 describes the nine tiers as they are defined by JMNO.

The area of interest for JMNO resides at tiers seven and eight, where the lower level military units can benefit from improved networking solutions. Desirable solutions should avoid any changes to the higher tier network structures. This thesis seeks to remain within these constraints as much as possible. It will not refrain, however, from exploring solutions that might impact other tiers when those solutions provide the best and most simple networking approach.

Tier	Components
Tier 0	Defense Video Services – Global (DVS-G) Defense Switched Network (DSN) Defense Message System (DMS) Transition Hub (DTH) Defense Red Switch Network (DRSN) Unclassified but Sensitive Internet Protocol Router Network (NIPRNET) Secret Internet Protocol Router Network (SIPRNET) Joint Worldwide Intelligence Communications System (JWICS) Trojan Spirit
Tier 1	Defense Information System Network (DISN) long-haul system
Tier 2	Standardized Tactical Entry Points (STEP) Teleports
Tier 3	Theater Resources of the Regional Combatant Commander Theater Headquarters Theater Network Operations Control Center
Tier 4	Joint Task Force (JTF) Headquarters Service Component Headquarters – Army Forces (ARFOR), Marine Corps Forces (MARFOR), Navy Forces (NAVFOR), and Air Force Forces (AFFOR) Headquarters
Tier 5	Army Corps Marine Expeditionary Forces (MEFs) Navy Carrier Battle Groups (CVBGs) Amphibious Ready Groups (ARGs) Numbered Air Forces
Tier 6	Divisions, Wings, and Naval Task Forces
Tier 7	Brigades, Regiments, Groups, and Task Units
Tier 8	Battalions, Squadrons, and Ships

Table 1. JMNO Network Tiers.

3. JMNO Autonomous System Numbering Standards

In order to utilize Border Gateway Protocol (BGP), each Autonomous System (AS) must be assigned a unique number. With the large number of BGP implementations worldwide, however, the availability of globally unique numbers is rapidly being depleted. A solution to this problem is the use of private AS numbers. Many networks within an isolated area can use private AS numbers that are not seen by the Internet at large. The range of private BGP AS numbers is 64512-65535. When using private AS numbers, the lower level organizations are effectively hidden from the Internet at large because all connections go through a high level entry point. By default, private AS numbers are not advertised outside of their stub network area. The higher level units have official, registered AS numbers that are authorized for “external-world” connectivity.

JMNO has developed a standard approach for use of certain private AS numbers by the military branches. This approach provides each Service with a range of AS numbers that can be allocated within the military network. These standard numbering assignments are shown in Table 2.

AS Number Range	Service Branch
65010 – 65019	U. S. Army
65020 – 65029	U. S. Marine Corps
65030 – 65039	U. S. Navy
65040 – 65049	U. S. Air Force

Table 2. JMNO AS Number Assignments.

JMNO has selected a relatively small range of AS numbers for these allocations based on standard military network configurations. Should an individual Service run out of AS numbers, it is possible to draw additional unused AS numbers from the private numbering range. This would also have to be done

for any other governmental agencies, non-governmental agencies, or foreign militaries that are allowed to connect to the network. The senior network operations center in the operational area should coordinate deconfliction of these additional assignments. As a recommendation from this thesis, it seems more efficient to allocate larger ranges of AS numbers to each Service. Instead of blocks of ten addresses, each Service could be given a block of one hundred addresses. This would allow for growth and flexibility in the BGP network. This thesis, however, uses the standards provided by JMNO in Table 2.

B. RESEARCH OBJECTIVES

The JMNO solution seeks to reduce reliance on independent links and to increase the sharing of resources among the Services. The desired effect will be to increase cross-Service communications without relying on each branch having to reach to the highest level before data can be exchanged with other Service networks. Other JMNO research is being conducted to develop use case scenarios, define information assurance procedures, and determine appropriate policy guidance. This thesis was started with a core goal of providing Quality of Service (QoS) guarantees to allow the mobile user to accomplish its mission without allowing the host Service's network to be overwhelmed. The goals have grown as research into the problem area revealed much more work needed in developing network routing solutions. This thesis, therefore, has evolved in scope to include both network routing and QoS solutions. The culmination of research for this thesis was manifested in a network test-bed that demonstrates routing and QoS solutions on a simulated tactical network backbone. The criteria for evaluating solutions include: ease of configuration, scalability, operational effectiveness, and simplicity.

Network routing is a critical element of the JMNO solution. Lateral connections between Service nodes should provide improved information flow without relying on connections through higher command nodes. Additionally, mobile units should be able to connect via any available Service node and

achieve connectivity to its parent command. It is not desired, however, that all of its traffic flow through the parent command. If it needs to reach an external site, such as a SIPRNET weather portal, it would be preferred that the traffic be routed through the connecting Service network without the added network burden of reaching back to its parent command first.

The QoS solutions examine control implementations that permit mobile users to gain access through the networks of other Services while assuring that host Service of a certain level of network performance. It also seeks viable methods for negotiating the competition for resources between the units involved. The thesis attempts to demonstrate that through the employment of differentiated services, it may be possible to provide QoS controls without knowing beforehand the exact bandwidth requirements of each user.

C. RESEARCH QUESTIONS

1. What is the typical network topology into which JMNO users will be connecting?

2. What JMNO tactics, techniques, and procedures (TTP) information is required to plan and implement QoS controls for JMNO users?

3. Are there any previous QoS implementations or designs that might be applicable to JMNO, such as between commercial network providers?

4. How can the bandwidth allocation for JMNO users be adapted according to the local network load?

5. Can Differentiated Services be used to provide separate classes of JMNO users connecting through the network? If so, what per-hop-behaviors (PHBs) are suitable?

D. ORGANIZATION

The remainder of this thesis follows the structure described below.

Chapter II provides a discussion of the primary networking and QoS building blocks that are used in the research. Networking protocols that are discussed include Enhanced Interior Gateway Routing Protocol (EIGRP), Border Gateway Protocol (BGP), and Dynamic Host Configuration Protocol (DHCP). After a discussion of QoS principles, further discussion explores the possible approaches that were considered for implementation. These include the Best Effort Service, Integrated Service, and Differentiated Service models.

The proposed network routing solution is described in Chapter III. It provides detailed information regarding the configuration of both external and internal BGP. This chapter also proposes a DHCP solution for managing address assignments for mobile units. It concludes by describing the required steps for a mobile unit to connect after moving to another Service's area of operations.

Chapter IV provides a discussion of the proposed QoS solution. It discusses the implementation of Differentiated Services. This includes traffic characterization, class definition, and policy mapping. This chapter provides detailed information regarding an actual QoS solution being implemented on an operational network.

Chapter V discusses recommendations for JMNO use of this research and provides some potential areas for future work. The Appendices include the full router configuration files for all nodes in the final laboratory network.

II. BACKGROUND AND RELATED WORK

A. ROUTING DESIGN

Providing network services for JMNO users relies on establishing a backbone router network that allows traffic to move effectively between nodes. The design and configuration of the network routers significantly affects the usefulness of routed application services. Users desire speed, reliability, and simplicity from the network. The actual implementation of these characteristics requires the use of various router configurations and protocols.

This thesis seeks to implement various protocols in ways that meet the specific requirements of JMNO. An explanation of some of the critical protocols under consideration is warranted to understand why a particular protocol is used or not used. Protocols provide a standard means of interaction between networking devices. In order to achieve a level of standardization and compatibility, a new protocol is generally published through the Internet Society in the form of Request for Comments (RFC). This allows experts in the field to review the protocol and to provide input where appropriate. When the protocol is deemed suitable for adoption by the Internet Engineering Task Force (IETF), it can be established as an Internet standard [2]. It is clearly important that JMNO use shared protocols between operational units for effective communication to take place. Some key protocols for designing and implementing a routing structure for JMNO are discussed in the following sections.

1. Enhanced Interior Gateway Routing Protocol

Enhanced Interior Gateway Routing Protocol (EIGRP) is a routing protocol developed for proprietary use on Cisco routers. Because the military Services currently employ Cisco routers exclusively, many have adopted EIGRP for standard usage on their networks. Because of this, JMNO has determined that EIGRP is a viable choice for routing solutions and has actively sought solutions

that incorporate this protocol – for both internal and external routing [1]. This thesis does not support the use of EIGRP for external routing. It assumes the position that internal routing should be determined independently by the military Service branches. An examination of EIGRP is provided here to understand the protocol in order to make informed decisions regarding its employment.

Because EIGRP is a proprietary solution, it is not defined within a publicly available RFC. Instead, details about the operation and use of EIGRP are found in Cisco documentation [3]. Cisco advertises EIGRP as an interior gateway protocol that "scales well and provides extremely quick convergence times with minimal network traffic [3]." EIGRP is a distance vector routing protocol; it builds its topology table by storing all advertisements from its neighbors and then converges the data by searching for a candidate loop-free route. If it does not know a route, it will query its neighbors. When it finds multiple routes to a destination, EIGRP stores all of the routes but uses the one that it selects as the best path. If a link fails, EIGRP is able to immediately begin using alternative routes that are already stored in its topology table.

a. Neighbor Discovery

EIGRP distributes network routing information by sending non-periodic updates. It only sends routing updates when paths change. The routes themselves do not time out. This could cause a situation where a route through a neighbor is not operational but the home router does not know about this. To account for this, EIGRP relies on neighbor relationships to distribute routing table information. The neighbor relationship is dependent on an exchange of hello packets.

Two routers running EIGRP will send hello packets when they encounter each other on the same network. This establishes the neighbor relationship between the routers. They then continue to exchange hello packets periodically to maintain the relationship. The frequency of these exchanges, the "hello-interval," depends on the speed of the link between the routers. For high

bandwidth links, EIGRP will send hello packets every five seconds. EIGRP considers broadcast media (like Ethernet), point-to-point serial links, and multi-point circuits with speeds greater than T1 (1.544 Mbps) to be high bandwidth links. For lower bandwidth links, EIGRP sends hello packets every 60 seconds. It considers multi-point circuits with speeds of T1 or less to be lower bandwidth links. The interval between hello packets can be adjusted on the router through the “ip hello-interval eigrp <custom-value>” command. The router also has a parameter called “hold-time” that determines the amount of time it will allow between hello packets from its neighbors before deciding that the link is down. Generally, the hold-time is set for three times the hello-interval. The standard hold-times are 15 seconds for high bandwidth links and 180 seconds for low bandwidth links. The hold time can be customized through the router with the “ip hold-time eigrp <custom-value>” command. When the hello interval is manually configured, the hold-time does not automatically adjust, so it would need to be manually set, as well.

EIGRP only builds neighbor relationships based on the primary IP address of the connecting interface. It does not use secondary addresses to determine neighbor relationships. EIGRP has no inherent limit to the number of neighbors that it can support. Instead, the number of neighbors is limited by the capability of the router itself. This limit is determined by the memory capacity, processing power, amount of information exchanged (e.g. the number of routes sent), topology complexity, and network stability.

b. EIGRP Topology Table

EIGRP does not rely on its routing table to store all of the information it needs to perform routing functions. Instead, it builds a second table to hold the required information. This table is called the topology table. EIGRP then uses the topology table to provide the information required to build its routing table. A record is built in the topology table for each neighbor. Each

record contains specific information about the known paths. Information stored in the records of the topology table includes:

- Lowest bandwidth on the path to the destination as reported by the upstream neighbor.
- Total delay.
- Path reliability.
- Path loading.
- Minimum value of the maximum transmission unit (MTU) along the path.
- Feasible distance.
- Reported distance.
- Route source [3].

c. EIGRP Metrics

To select routes from the topology table, EIGRP uses two factors: minimum bandwidth on the path to the destination network and total delay. Although it is possible to configure other metrics to perform this function, Cisco does not recommend doing this due to the possibility of creating routing loops within the network [3]. The bandwidth and delay metrics are determined from the values configured on the interfaces along the path to the destination network. To scale the bandwidth number, EIGRP uses the formula: bandwidth = (10,000,000 / bandwidth(i)) * 256, where bandwidth(i) is the minimum bandwidth on all outgoing interfaces along the path in kilobits. Thus, the bandwidth metric is effectively normalized, inversely, to the Ethernet bandwidth. To scale the delay number for inclusion in the overall metric determination, EIGRP uses the formula: delay = delay(i) * 256, where delay(i) is the sum of delays configured on interfaces along the path in tens of microseconds [3]. For a simple, default metric determination EIGRP then simply adds the bandwidth and delay values. It is possible to compute the metric differently, but for brevity that will not be discussed here. This final metric is then compared with other routes to the same destination and the route with the smallest metric value is selected.

d. Configuring EIGRP

The ease of configuring EIGRP on Cisco routers is one of the primary benefits of its use. In a very basic implementation a router is configured to run EIGRP by assigning it an EIGRP Autonomous System (AS) Number and telling it which network(s) to include. There is no relationship between EIGRP and BGP AS numbers. Because the neighbors automatically find one another and exchange topology table information, the user does not need to configure all of the details involved in the protocol. This basic configuration creates an EIGRP routing process and does not necessarily require any additional action to get the protocol operating. All other EIGRP configuration settings are considered optional, with creating the routing process being the only mandatory procedure. The commands used to create the EIGRP routing process are shown in Table 3.

Command	Purpose
Router(config)# router eigrp <*autonomous-system*> Example: M1(config)# router eigrp 65021	Enables an EIGRP routing process
Router(config-router)# network <*network-number*> [*network-mask*] Examples: M1(config-router)# network 192.168.68.0 M1(config-router)# network 192.168.64.0 0.0.31.255	Associates network(s) with an EIGRP routing process

Table 3. Creating an EIGRP Routing Process.

In addition to the basic EIGRP configuration, it is often desirable to incorporate some additional properties. One common addition that is used is to redistribute routes into EIGRP from other routing processes. The EIGRP autonomous system needs to learn about neighboring networks (i.e. networks that are not part of the local EIGRP AS) to enable routing to these external

locations. The neighboring network might be running another implementation of EIGRP or a completely different routing protocol. In either case, the local network will need to learn about the networks that can be reached through that protocol's routing information base so that it can send traffic there as appropriate. An example of redistribution between two EIGRP neighbor processes is shown in Figure 1. In this example, Router A is operating in EIGRP AS 32 and Router C is in EIGRP AS 64. Router B connects the two ASs and provides the redistribution between the EIGRP routing processes. The use of route maps allows Router B to ensure it does not forward routes learned from one network back to the same network after redistribution. For example, if Router C knows of a different network it could pass that information to Router B through EIGRP 64. During the redistribution, Router B would tag that route with 64 to know where it was learned. It will then pass the information to Router A through EIGRP 32. When Router A learns this route it adds it to its routing table and will forward it and all the routes it knows back to Router B. Because of the route map, Router B will block any routes with the 64 tag from being pushed back into EIGRP 64.

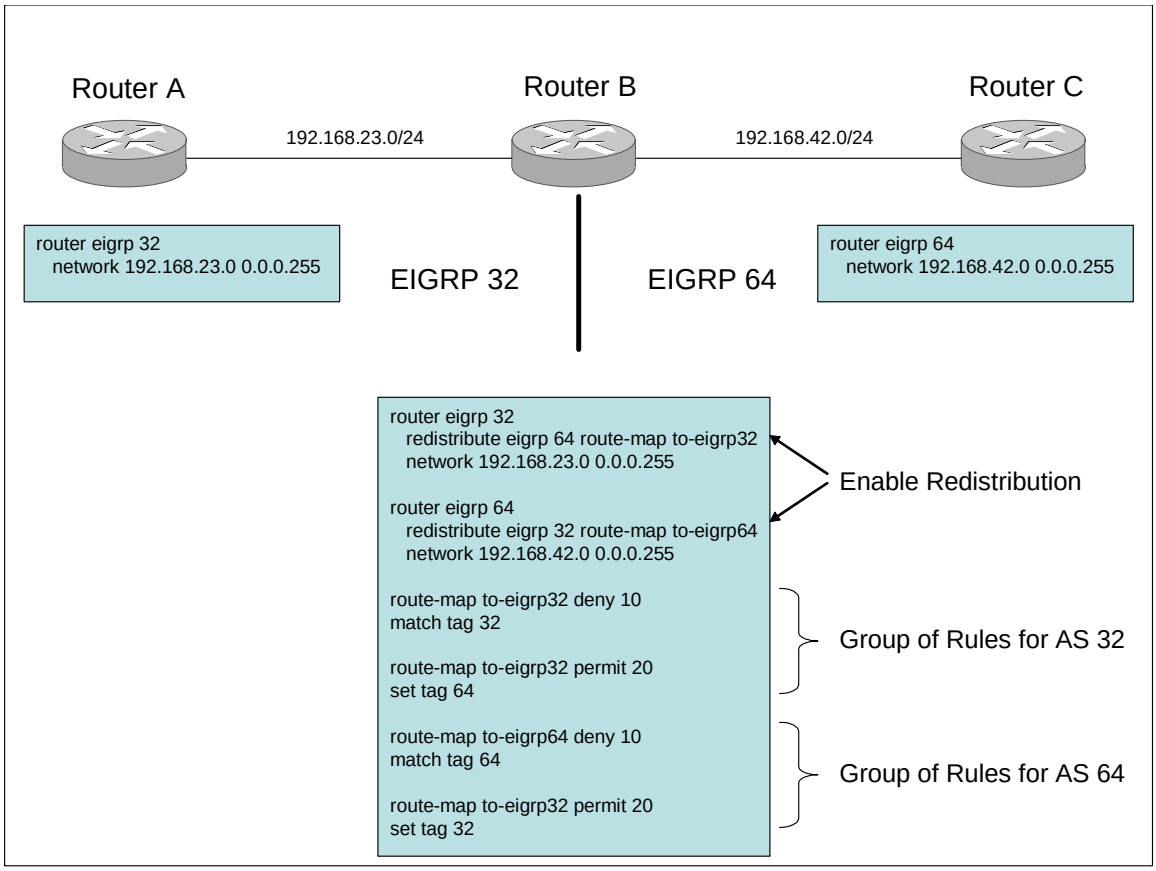

Figure 1. Redistributing EIGRP to EIGRP.

2. Border Gateway Protocol

Although EIGRP is a useful protocol for providing internal routing, it was not designed to connect large networks together. For joint operations, where many military networks might be interconnected, it seems as though there must be a more appropriate approach. This is even more critical when considering the other government agencies, non-governmental agencies, and foreign militaries that might also be connected. The standard solution used throughout the Internet is the Border Gateway Protocol (BGP) because it provides a decentralized, scalable approach to routing. BGP provides the routing protocol for connecting distinct autonomous systems. As defined by [4], BGP is currently

on version four, but according to standard community practices it will only be referred to as BGP (vice the more specific BGP-4).

BGP provides policy-based routing options, rather than simple distance vector routing. This permits each AS to develop its own policies regarding how it wants to route traffic through its neighbors. Depending on the implemented policies, BGP allows an AS to learn about subnet reachability from neighbors, send this reachability information to all internal routers within the AS, and determine which routes to use based on reachability and AS policy. The following Sections will discuss some key attributes and configuration requirements for BGP.

a. BGP Sessions

For BGP to function, two neighboring routers must communicate with one another so they can learn about available routes. This communication is done through a Transmission Control Protocol (TCP) connection between the routers. The TCP connection and all associated messages sent over it are considered a BGP session. If the session is between neighbors in different ASs, it is considered an external BGP (eBGP) session. When the two routers belong to the same AS, the session is called an internal BGP (iBGP) session. A depiction of the finite state machine showing the BGP session with possible event transitions as described in [4] is shown in Figure 2. This depiction does not provide any additional actions that BGP performs during the transitions, such as restarting timers or releasing resources.

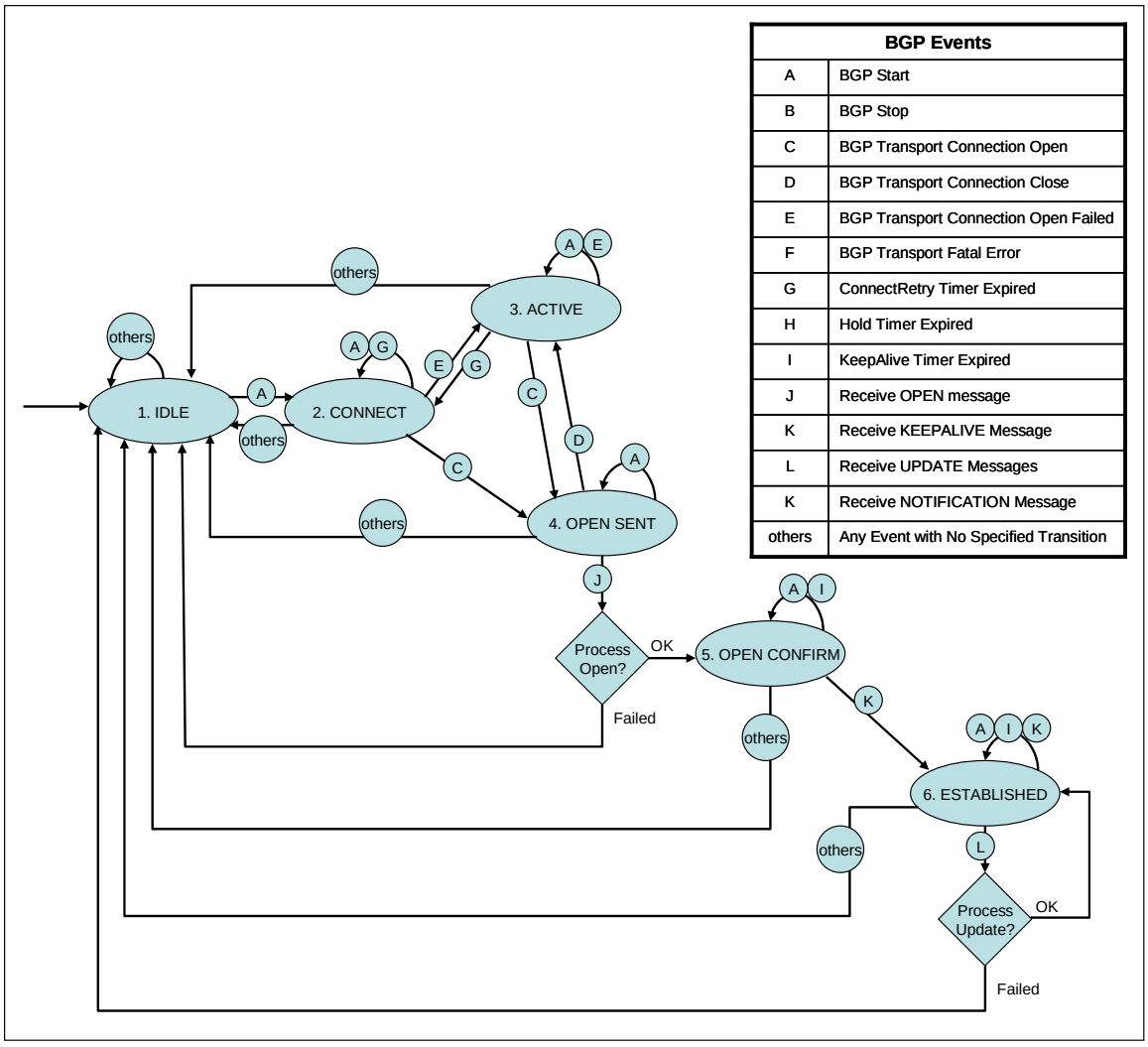

Figure 2. BGP Finite State Machine.

b. BGP Path Attributes

BGP takes a path vector approach to routing. Routers advertise the routes they know for reaching destination networks. A router will pass not just the network address, but also key attribute information. This additional information allows other routers to update their routing tables and to select routes. Three of the most important attributes are local preference, AS path, and next hop [5].

(1) Local Preference. BGP allows policies to indicate which routes it prefers to use. This policy decision can be made by the network administrator or learned from another router within the same AS. The local preference value is a four octet non-negative integer contained in the LOCAL_PREF field. Larger LOCAL_PREF values have higher precedence than lower ones.

(2) AS Path. The path to a particular destination network is found in the AS_PATH attribute. The AS_PATH contains the number of each AS that the route has been sent through. Each AS will append its own AS number to this field before advertising the route to another AS. Routers can use the AS_PATH field to ensure that it does not add loops to their routing tables. This field can also be used to select routes to particular networks.

(3) Next Hop. The NEXT_HOP attribute contains the IP address of the interface to the next AS gateway on the path to a destination network. This allows routers to make decisions when they may have learned multiple paths to a destination. Because an AS might have multiple inbound and outbound links, a particular router might learn of different routes to the same destination from eBGP and iBGP neighbors. The NEXT_HOP attribute allows the router to make a decision based on the next router on the path.

c. BGP Route Selection

When multiple routes to a given destination are received, a router must be able to choose one to use. By using the attributes discussed above, BGP provides a mechanism for selecting routes. The first criterion applied is the LOCAL_PREF attribute. BGP will examine the potential routes and select the one that has the greatest LOCAL_PREF value. If there are multiple routes sharing this value, then BGP next looks at the AS_PATH attribute. It will select the route with the smallest number of AS hops to the destination. This does not consider the number of router hops that might be encountered within a particular AS. Next, BGP will consider the NEXT_HOP attribute when there is still a tie

between potential routes. It considers the closest NEXT_HOP router to be the one with the least-cost path to reach it. The least-cost path is determined by the interior routing protocol in the AS. If there is still a tie among the routes, the BGP will apply other attributes, such as how the network was learned by BGP, until only one route remains [5].

d. BGP Routing Policies

BGP AS administrators might want to control which traffic is allowed to transit its nodes. For commercial Internet Service Providers (ISPs), this is generally desired for fiscal reasons. In both private and military networks, however, there may also be operational goals that drive these policies. One of the JMNO goals is to ensure that lateral links provide improved network performance to tier seven and eight units. If these links are advertised to higher tier units, the links might experience congestion and reduce the benefit of lateral connectivity to the lower tier units. An example where the AS administrator might not want to allow all traffic to use certain links is shown in Figure 3. While ARMY_BGDE desires to provide enhanced connectivity between lateral units, it does not want to have higher unit traffic causing congestion. Specifically, ARMY_BGDE might not want to route MARFOR traffic to Air Force units. In this example, it would be preferred for MARFOR to send its traffic through the JTF network for this purpose. However, it might be desirable for the Marine Tactical Air Command Center (TACC) to be able to use the lateral links to reach the Air Force Control and Reporting Center (CRC). This would require cooperation and policy integration between the lateral units.

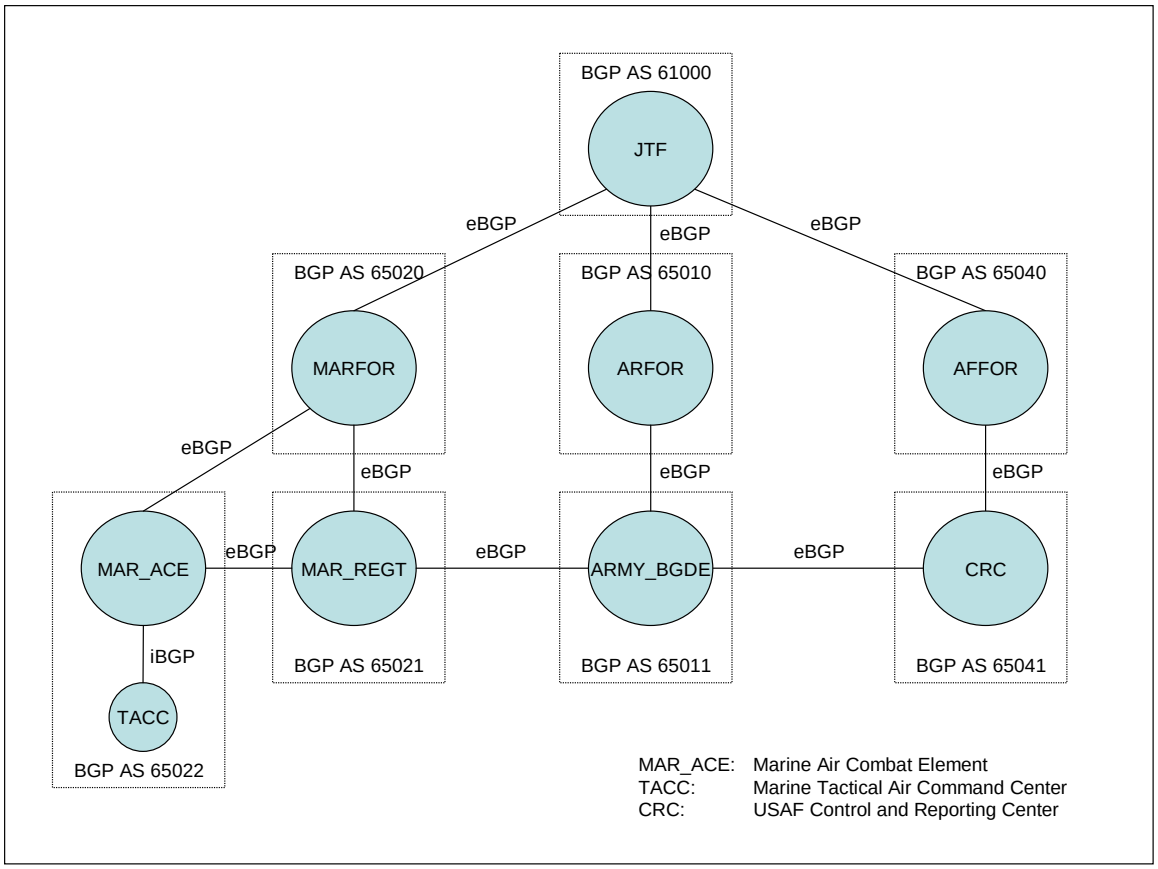

Figure 3. Notional BGP Scenario.

The use of policy filters can allow AS administrators to control which routes are advertised to particular neighbors [5]. Using the example from above, ARMY_BGDE might advertise lateral links to the other lateral neighbors but not to ARFOR. In this way, MAR_REGT could learn about routes to the CRC. With an agreement in place, MAR_REGT might advertise routes it learned from ARMY_BGDE to the Marine Air Combat Element (MAR_ACE) but not to MARFOR. MAR_ACE in turn would learn of the lateral link to the CRC, but would not advertise any routes it learned from MAR_REGT to MARFOR. Because the TACC is effectively a customer of MAR_ACE, it will be able to use the lateral link for coordination with the CRC.

e. Configuring BGP

BGP is an extremely complex protocol and not all of its details will be discussed in this thesis. However, some of the most common configuration settings should be examined. These include enabling BGP on the router, adding network statements, creating neighbor relationships, and creating path filters. All configuration examples provided are designed for Cisco routers since that is the hardware of choice for JMNO. Because BGP is an open standard, the solutions should easily translate to other hardware if other routers are incorporated into the network.

(1) Enabling BGP. Before any other BGP configuration settings can be set, an instance of the protocol must be established on the router. This is done by starting from the global configuration mode. The command is then entered with keywords “router bgp” followed by the autonomous system number [5]. This command automatically places the user into the router configuration mode. The format for enabling BGP on a Cisco router is shown in Table 4.

Command	**Purpose**
Router (config)# router bgp <*as-number*>	Enables a BGP routing process

Table 4. Enabling BGP.

(2) Adding Network Statements. BGP requires knowledge of which networks are local to its AS. This allows it to initialize its routing information base (RIB). In the absence of any other restrictions, the networks added to the BGP RIB will be advertised to all BGP neighbors. Starting in the router configuration mode, the keyword “network” is followed by the network address. This will include the network based on classful addressing space. Optionally, a network mask may be included to designate subnetting or supernetting. Another optional entry allows a route map designation to be

associated with the network [5]. After creating a network statement, the router remains in the router configuration mode. The format for adding network statements is included in Table 5.

Command	Purpose
Router(config-router)# network <*network-address*> [*network-mask*] [route-map <*route-map-name*>]	Adds network as local to AS

Table 5. Adding BGP Network Statements.

(3) Creating Neighbor Relationships. Neighbors in BGP must be defined explicitly to create the TCP connection and begin exchanging information. This is true for both eBGP and iBGP neighbors. Creating a neighbor relationship begins in the router configuration mode. The keyword "neighbor" is followed by either the IP address of the neighbor or its peer group number (if used). This is then followed by the keyword "remote-as" and the AS number of the neighbor [5]. For an iBGP neighbor, the AS number will be the same one as the one assigned to the router being configured.

At this point, the neighbor relationship is defined and does not require any additional commands. However, some additional commands may be implemented in defining neighbor relationships that can make BGP function more effectively. One useful optional command permits BGP to process and store policy updates without clearing the entire session. This command uses the "soft-reconfiguration inbound" keyword and is associated with individual neighbor relationships. Another optional configuration for BGP neighbors allows the peering relationship to be built on interfaces that are not directly connected. This is particularly useful for iBGP neighbors that might not be fully meshed with physical connections. This configuration is created by using the keyword "update-source" followed by the interface to be used for the relationship. A

common application is to use a loopback interface to define iBGP relationships. The format for the commands used to define neighbor relationships is depicted in Table 6.

Command	Purpose
Router(config-router)# neighbor {*ip-address* | *peer-group-number}* remote-as <*as-number*>	Adds a neighbor relationship
Router(config-router)# neighbor {*ip-address* | *peer-group-number}* soft-reconfiguration inbound	(Optional) Allows policy updates without rebuilding entire BGP process
Router(config-router)# neighbor {*ip-address* | *peer-group-number*} update-source <*interface*>	(Optional) Defines a particular interface for the BGP peering relationship

Table 6. Defining BGP Neighbor Relationships.

(4) Creating Path Filters. BGP does not require path filters to be defined. This optional configuration can be very useful, though, when network policies desire control over route advertisements. The first step in creating path filters is to create an access list based on AS path values. The command begins in global configuration mode with keywords “ip as-path access-list” followed by a user-defined list number. This is then followed by either keyword “permit” or “deny” and a regular expression to match the desired AS number attribute [5]. Multiple access lists and/or regular expressions may be defined in separate statements. The next step is to enter the router’s BGP router configuration mode. From there, the filter list is then applied to individual neighbors for either inbound or outbound routes. The command begins with keyword “neighbor” followed by the IP address or peer group name of the neighbor. Then the keyword “filter-list” is followed by the previously defined list number and either keyword “in” or “out.” The command format for creating path filters in BGP is depicted in Table 7.

Command	Purpose
Router(config)# ip as-path access-list <*access-list-number*> {permit | deny} <*as-regular-expression*>	Defines a BGP related access list
Router(config)# router bgp <*as-number*>	Enter router configuration mode
Router(config-router)# neighbor {*ip-address* | *peer-group-name*} filter-list <*access-list-number*> {in | out}	Establish a BGP path filter

Table 7. Creating BGP Path Filters.

3. Dynamic Host Configuration Protocol

Assignment and management of internet protocol (IP) addresses within networks can be a cumbersome task if done manually. As mobile network devices move from one network to another, this task becomes even more challenging. For JMNO situations where mobility of users and units is a critical issue, finding methods to simplify the management of IP assignments is clearly important. One means of simplifying this process is through the use of Dynamic Host Configuration Protocol (DHCP). Defined in [6], DHCP allows clients to be allocated IP addresses and network configuration information automatically. Some of the key design goals of DHCP are listed below.

- Clients should not require manual configuration.
- Networks should not require manual configuration for individual clients.
- DHCP should not require a server on each subnet.
- DHCP must coexist with statically configured, non-participating hosts, and with existing network protocol implementations.
- Guarantee that any specific network address will not be in use by more than one client at any time [6].

a. DHCP Operation

DHCP operates through a client-server relationship. The client is the device requiring an IP address. The server runs DHCP to allow it to provide network information to clients. If no server is operating on the subnet to which the client connects, a DHCP relay agent can forward the information to the DHCP server. The actual process for a client to negotiate an IP assignment requires four steps.

(1) DHCP Server Discovery. The initial contact made by the client is seemingly problematic. It does not know the address of the DHCP server, does not know the subnet to which it is attached, and does not have an IP address for itself. To accomplish this task, the client first creates a DHCPDISCOVER message within a UDP packet. This message will contain a transaction ID to allow responses to be matched appropriately. It then creates an IP datagram encapsulating that message with a broadcast destination address 255.255.255.255 and a source address of 0.0.0.0. This datagram is then encapsulated into a link-layer frame containing the client's MAC address and the broadcast MAC address FF-FF-FF-FF-FF-FF. This frame is then sent on the network segment. If a DHCP server is running on the segment, it will process the message. Otherwise, a DHCP relay agent can forward the message to the network that contains a DHCP server.

(2) DHCP Server Offer. After the server receives the DHCPDISCOVER message, it determines whether or not it can support the request. If it does not have an address available, the server may provide notification to the system administrator [6]. If it does have an available address, it will respond to the client with a DHCPOFFER message. This message will contain the original transaction ID, an IP address, the network mask, and the length of time for which the lease will be valid.

(3) DHCP Request. A client might send out its initial DHCPDISCOVER message and find that it receives multiple DHCPOFFERs. This is possible because there may be more than one DHCP server running on the network and each responds with its own offer. The client, then, will select one of the offers it receives. It will then send a DHCPREQUEST message that contains all of the information from the offer. The DHCPREQUEST is broadcast to all servers to inform them that an offer has been accepted.

(4) DHCP Acknowledgment. The final step in the process is for the DHCP server to respond to the DHCPREQUEST. It does this by sending a DHCPACK message back to the client confirming the assignment and all of the corresponding parameters. The overall flow of the DHCP assignment process, with one server, is depicted in Figure 4.

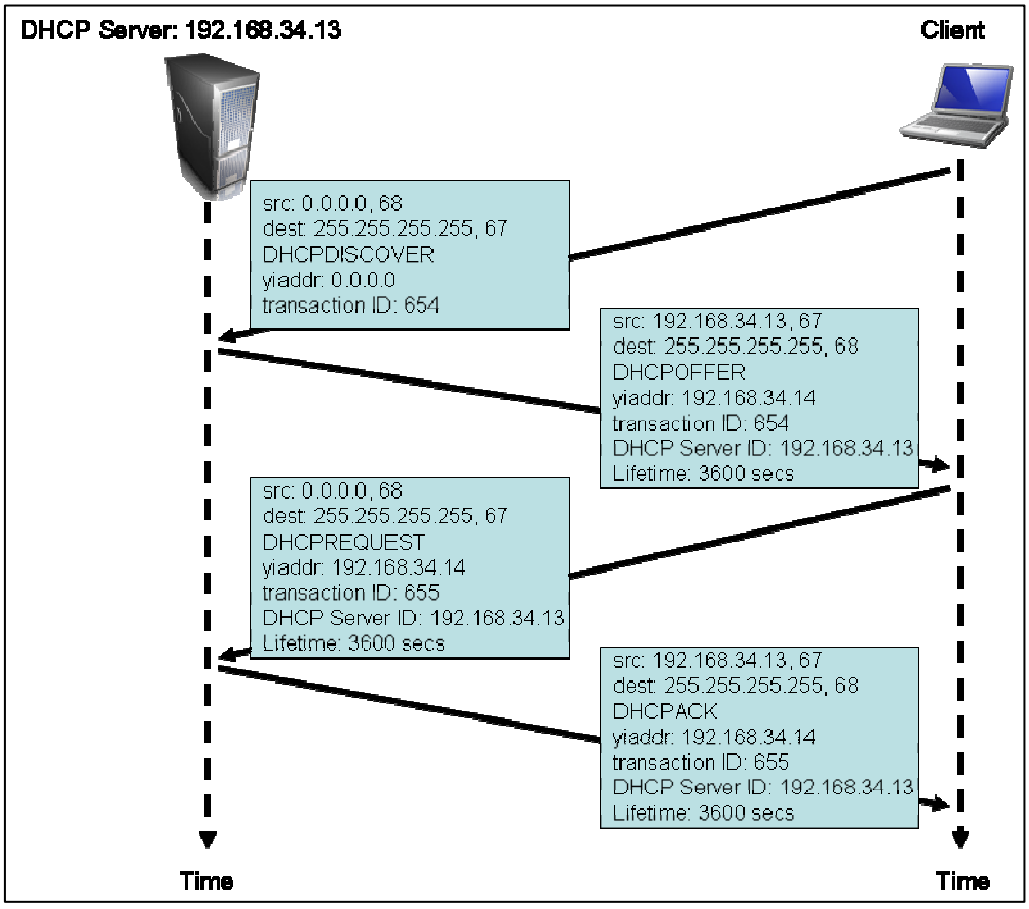

Figure 4. DHCP Client-Server Negotiation (After: [2]).

b. DHCP on Cisco Routers

The use of DHCP for client devices is common on many networks. It is less likely, however, to be encountered on network routers. The RFC actually states that DHCP is not intended for use in configuring routers [6]. There may be times, though, where an edge router could benefit from the automated IP assignment process for its network interface. Cisco routers have been designed to allow interfaces to receive IP addresses through DHCP. The routers are also capable of being configured as DHCP servers, if desired. The utility of this type of function is that mobile routers might be able to connect to a DHCP server-enabled router and receive its IP assignment without user configuration changes.

(1) DHCP Server Implementation. In order to create the DHCP service on a router, a pool of IP addresses must be known and available for use. On Cisco routers the pool is defined through the configure-terminal mode with the command “ip dhcp pool [name],” where [name] is replaced with a desired string for that address pool. From this command, the user is able to input network addresses that should be included. This is accomplished by entering “network” followed by the network ID and the network mask. The network mask allows the administrator to constrain the address pool to a reasonable sized set of addresses, according to the expected number of simultaneous clients. From these addresses, the router can be assigned an IP for the interface through which clients are expected to connect. This interface’s IP address will become the default router for all clients using that DHCP pool. This is entered simply through the command “default-router” followed by the IP address. The router should then have that IP address excluded from the assignable addresses it will maintain. The command to do this is “ip dhcp excluded-address” followed by the IP address. Additionally, the actual interface must be configured as well by going into the interface configuration with the command “interface” followed by the name (Ethernet0/0, FastEthernet1/0, etc.). Then the address is assigned by using the command “ip address” followed by the IP address and the network mask. An example of these commands is provided in Table 8.

Configuration Input	Action
ip dhcp pool Router1 network 192.168.71.0 255.255.255.0 default-router 192.168.71.1	Create DHCP Pool for Assignment
ip dhcp excluded-address 192.168.71.1	Exclude Server Interface IP Addresses
interface Ethernet1/0 ip address 192.168.71.1 255.255.255.0	Assign IP Address to Outbound Interface

Table 8. Sample Router Configuration as DHCP Server.

(2) DHCP Client Implementation. The Cisco router implementation for allowing an interface to receive its IP address from a DHCP server is quite simple. It must be noted, however, that clients that wish to apply this technique to more than one interface "must use DHCP through each interface independently to obtain configuration information parameters for those separate interfaces [6]." Each interface is required to negotiate its own address assignment since they are most likely on separate network segments. To enable DHCP on an interface, the router must be configured through the global configuration mode. Then it must enter the interface configuration mode by the command "interface" followed by the interface number. For the IP assignment the user simply needs to enter "ip address dhcp." There are two additional fields that are optional for use but that do affect the contents of the DHCPDISCOVER packet. The first is the client-id field. This is used to tell the DHCP server which interface on the client is making the request. If the client-id field is not used, the default value sent by a Cisco router is the MAC address of that interface. The other optional entry is the hostname field. This is used to identify the name of the router initiating the request. If the hostname field is not used, Cisco routers will automatically use the globally configured hostname value contained in the

configuration file. Examples of the client configuration and their impact on the DHCPDISCOVER message are given in Table 9.

Configuration Input	DHCPDISCOVER Contents
hostname Router2 interface Ethernet0/0 ip address dhcp	client-id: <MAC Address of Eth0/0> hostname: Router2
hostname Router2 interface Ethernet0/0 ip address dhcp client-id Ethernet0/0	client-id: Ethernet0/0 hostname: Router2
hostname Router2 interface Ethernet0/0 ip address dhcp hostname NPSRouter	client-id: <MAC Address of Eth0/0> hostname: NPSRouter
hostname Router2 interface Ethernet0/0 ip address dhcp client-id Ethernet0/0 hostname NPSRouter	client-id: <MAC Address of Eth0/0> hostname: NPSRouter

Table 9. Sample Router Configuration as DHCP Client.

B. QUALITY OF SERVICE

In addition to network routing issues, JMNO is interested in providing different levels of network service to different traffic flows. For example, if the force commander is conducting an important video teleconference (VTC) with his subordinate commanders, it probably will be desired that the traffic involved in conducting this service receive higher priority than other flows. If the network involved had large enough bandwidth that it never experienced congestion, there would be no problem. All of the desired traffic would rapidly traverse the network and reach its destination. In a real network, however, this is very unlikely to occur. In a tactical military environment, where transmission link bandwidths are

often severely limited, the likelihood of no congestion becomes even smaller. A solution to this problem is the concept of Quality of Service (QoS).

QoS is the ability of a network to provide improved service to selected network traffic. The QoS solution can include many key features that improve the network service and make it more predictable. These features include: supporting dedicated bandwidth, improving loss characteristics, avoiding and managing network congestion, shaping network traffic, and setting traffic priorities across the network [7]. This Section will examine QoS fundamentals by examining some QoS principles and then looking at some methods used to support QoS in networks.

1. QoS Principles

Certain key principles are required in order to provide a QoS solution [2]. The first of these is that packets must be annotated in some manner. This classification allows routers to differentiate between packets belonging to different classes of traffic. This can allow for time-sensitive or critical traffic flows to receive priority treatment based on network policy. The packet marking does not provide any guarantee that it will receive a specific QoS, but it provides a means for routers to distinguish classes of traffic that they may then handle according to their policy configurations.

The second principle requires that traffic flows be provided some degree of isolation from other flows. This is required because a flow might misbehave and take more bandwidth than it has been allocated. This could cause other flows to starve, where their packets never receive service on the link because of the misbehaving flow. Hence, some mechanism is required to provide this isolation between flows to ensure all flows receive appropriate treatment from the network.

The third principle states that in providing isolation from other flows, it is desirable to use the network resources as efficiently as possible. If the network

were to divide bandwidth on links and reserve a portion for each flow, it is possible that some of the flows might not use all of their allocation. This would mean that other flows might still be restricted while part of the bandwidth remained unused. It is desired, therefore, that the isolation mechanism be able to ensure the network resources are used efficiently while performing its function.

The final principle requires that a call admission process be provided. This requires flows to declare their QoS requirements so that the network can determine whether to admit the flow if it can meet the requirements or to deny it access when the requirements cannot be met. This is important because the network may not have all of the resources available at the time of the request. If too many flows are admitted to the network, they will be competing for resources that are not able to handle all of their requirements simultaneously. This would defeat the QoS intent since obviously some of the flows would not receive the desired network handling. A call admission process handles this situation by ensuring that the resources are available before the flow is admitted to the network.

The four principles discussed are critical to understanding and implementing a QoS solution. However, some of the specific details of how those principles are put in practice are equally critical. The next two Sections will provide more details on the scheduling and policing mechanisms that are crucial to QoS.

a. QoS Scheduling

Data packets that arrive at a network node must be organized in some manner to be transmitted on the outbound link. To provide a QoS implementation, this scheduling mechanism plays an important role. Since it is desired to have certain packets receive priority handling, the scheduling mechanism should be capable of accommodating different classes of traffic. This section discusses several of the common methods of providing scheduling that are available for network implementation.

(1) First-In First-Out. The simplest scheduling mechanism is first-in first-out (FIFO). As the name indicates, packets are handled in the order they arrive at the node. When a packet arrives it is placed in a queue to wait for servicing. As the outbound link becomes available, the packet that has been in the queue the longest is selected for transmission. If the packets arrive at a faster rate than the outbound link can handle, the queue grows in length. If the queue becomes full, the node applies a discarding policy that drops certain packets. This policy can be set to drop the last packet to arrive or it can select some other packet to drop from the queue based on network policies. The operation of a FIFO queue is shown in Figure 5.

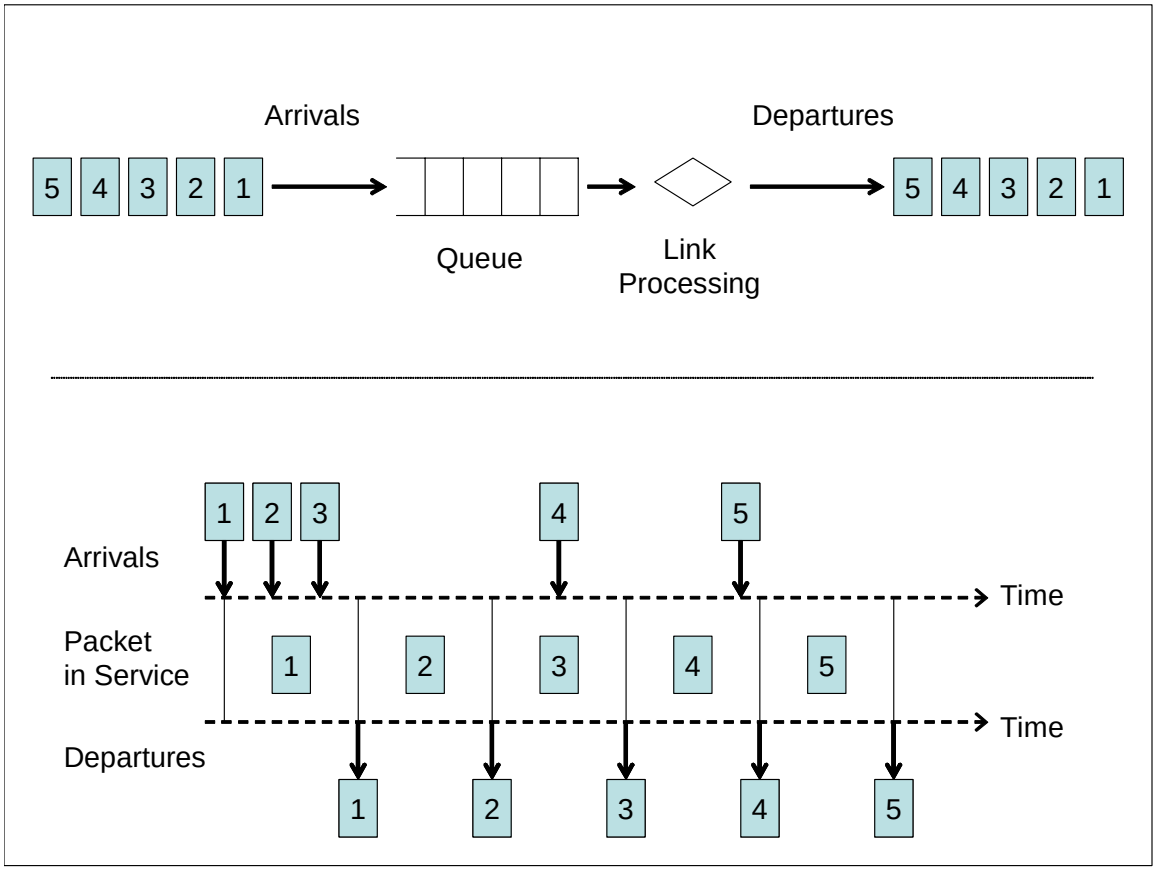

Figure 5. FIFO Queuing.

Most current internet routers operate simple FIFO scheduling processes. This is because it is the fastest method of scheduling and is very effective where links experience very little congestion. However, use of a standard FIFO process does not provide a means for providing QoS assurances. With a FIFO queue, “ill-behaved sources can consume all the bandwidth, bursty sources can cause traffic delays in time-sensitive or important traffic and important traffic can be dropped because less important traffic fills the queue [7].”

(2) Priority Queuing. Unlike FIFO procedures, priority queuing allows packets to be classified so that certain flows can receive expedited handling. As a packet arrives at the node, it is broken into a specific class of service based on policies. These classes can be determined by a variety of means, including the type of service (ToS) bits in an IPv4 header, the source or address, the source or destination port number, or other criteria. The node then has different queues to handle the different classes of traffic. Each queue is generally established as a FIFO queue. In a simple example there might be two queues, one for priority traffic, and another for standard traffic. When the outbound link is available for transmission it would first check to see if any packets were waiting in the priority queue and if so select the first of those for transmission. If the priority queue were empty, the node would then select the first packet waiting in the standard queue for transmission. A problem with priority queuing is that it allows certain flows to be denied access when the network is busy. In the example, if priority traffic flows arrived at a rate equal to or greater than the transmission rate, standard traffic would never be selected for transmission. A priority queuing mechanism and its operation is depicted in Figure 6.

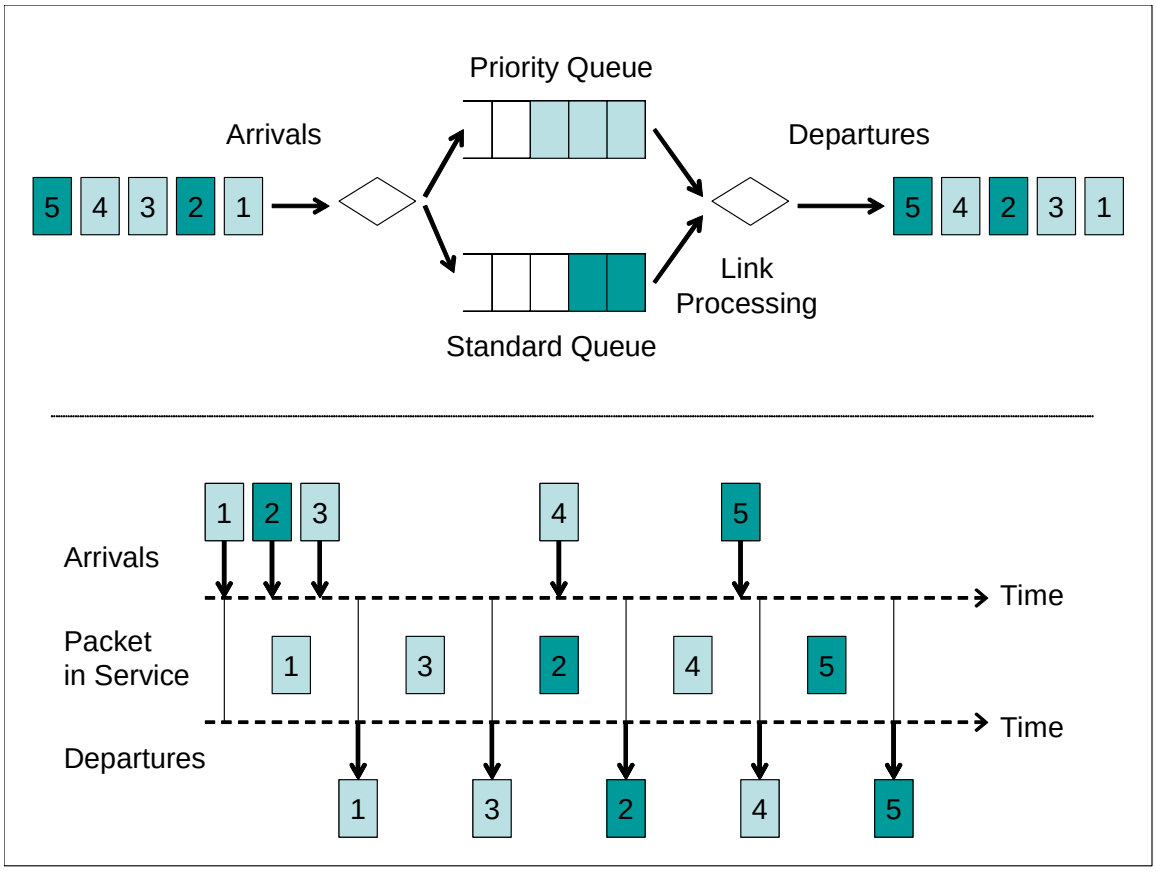

Figure 6. Priority Queuing.

(3) Round Robin Queuing. Round robin queuing attempts to account for the starvation problem of standard priority queuing. Again packets are organized into classes upon arrival and placed into appropriate class queues. The scheduler, however, does not rely solely on the class organization to select packets for transmission. Instead, it alternates its service among the classes to ensure that each class has an opportunity to receive service. If the round robin mechanism checks a class and there are no packets in the queue, it automatically moves on in its sequence to check the next class. An example of a round robin queuing mechanism is shown in Figure 7.

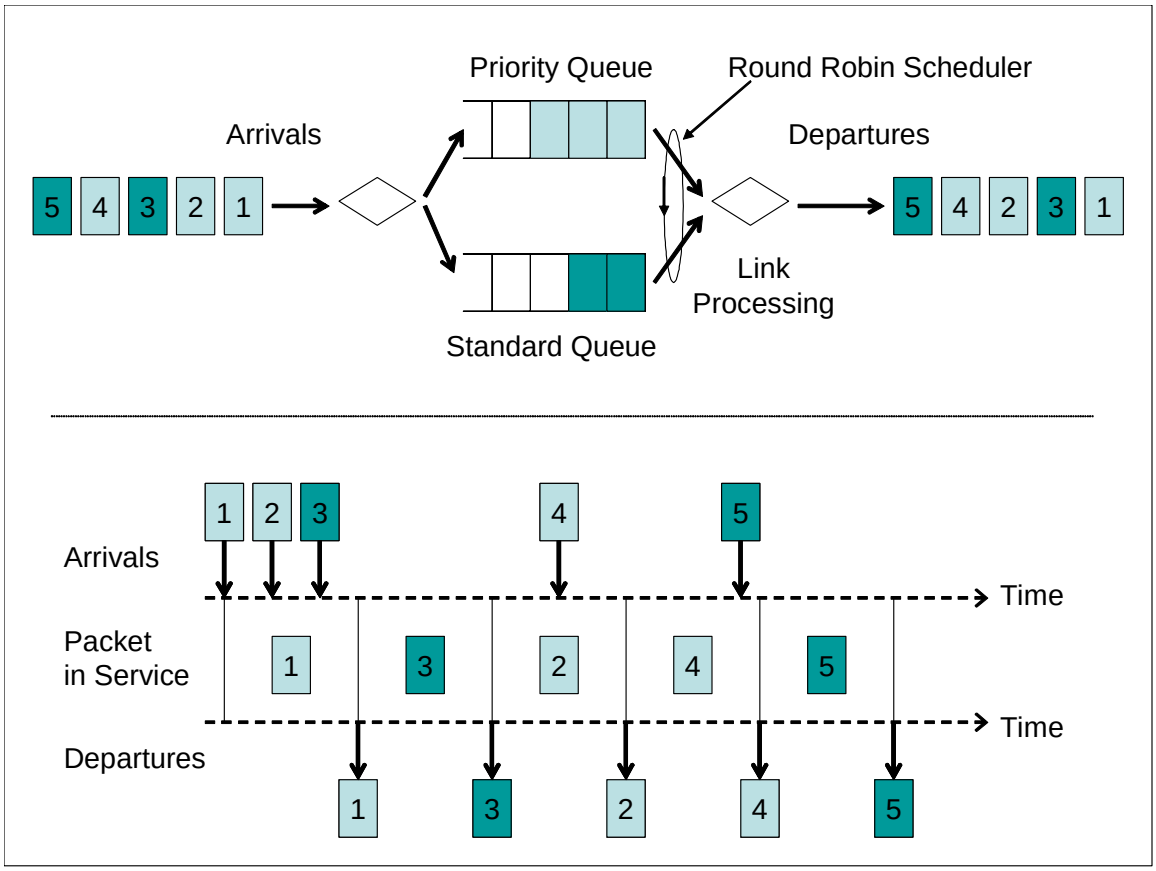

Figure 7. Round Robin Queuing.

(4) Weighted Fair Queuing. An adaptation of round robin queuing is weighted fair queuing (WFQ). In WFQ, incoming packets are organized into queues per class. The scheduler then moves through the classes in a sequential manner to select packets for transmission. Unlike round robin queuing, however, WFQ provides a guarantee that each class will receive a portion of the service time. Each class is assigned a weight that determines the amount of service it is guaranteed. Then each class is assured of receiving at minimum the portion of available service equal to its weight divided by the sum of the weights of all the classes. Because it is able to provide this type of guaranteed service, WFQ is often the preferred scheduling mechanism for QoS architectures [2]. An example of WFQ scheduling is shown in Figure 8.

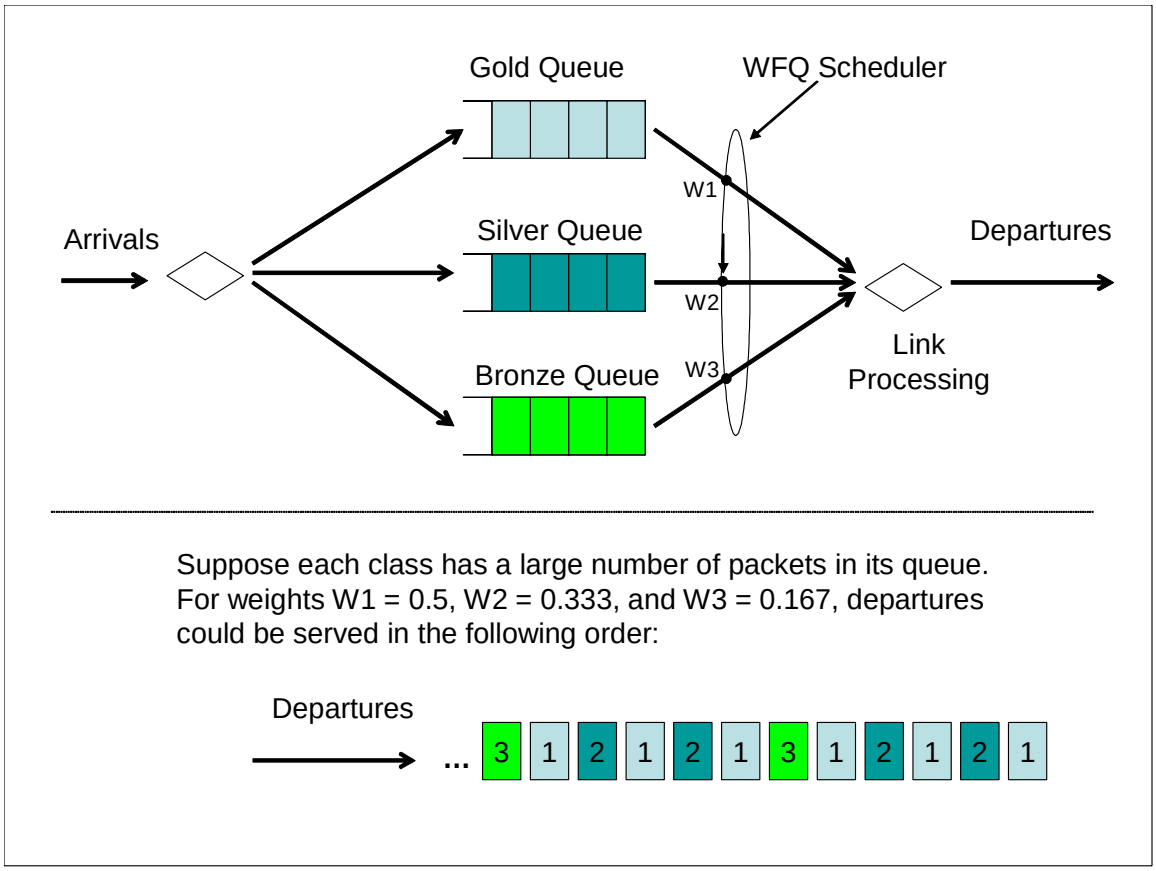

Figure 8. WFQ Scheduling.

b. QoS Policing

While scheduling handles the matter of selecting which flow to service at a given time, there is still a need to regulate how many packets a flow is allowed to push onto the network during a given interval. It is not desirable to have certain flows misbehave by injecting packets at a rate greater than allowed by network policies. To prevent this type of situation, the network needs a policing mechanism that can ensure compliance with network policies. This Section will look at the basic concepts involved in policing and the most common means of implementing a policing mechanism – the leaky bucket.

Before looking at the leaky bucket mechanism, it must be understood what traffic aspects need to be policed. There are three important

aspects of policing that must be considered. Each of these policing concepts plays a specific role in providing QoS on a network.

(1) Average Rate. The first policing concept is controlling the average rate at which a flow is allowed to send packets to the network over a period of time. Certain flows might attempt to use more than their share of the network's resources and cause other flows to suffer. Policing should be able to control the average rate that flows are able to send traffic. For example, the policing structure could limit a flow to sending no more than 3,000 packets per minute.

(2) Peak Rate. There is still a potential for misbehavior by traffic flows with controls on the average rate. Those controls limit rates over a relatively long period of time but might allow greater rates at smaller time intervals. In the example above, the traffic was limited to 3,000 packets per minute. An average rate policing structure might still allow this flow to send 1,500 packets in a one second interval within this limit. There is a danger that the peak rate could still send more packets than the buffer could handle, creating overflow. To protect against this type of behavior, a control on the peak rate must be implemented. Following the example, an additional limit might be to limit flows to sending no more than 100 packets per second.

(3) Burst Size. In addition to controlling average and peak rates, the network might want to provide limits on the amount of packets that can be sent in an extremely short period of time. Although a link cannot send multiple packets through a single interface at the same time, it is useful to consider the case where the interval between packets approaches zero. The burst size is a limit on the amount of packets that can be sent as this interval approaches zero. The policing mechanism might establish a limit of five packets, for example, for the amount of traffic that a flow can send at a virtually instantaneous time.

With an understanding of the basic policing concepts, it is possible to understand the functioning of the leaky bucket policy-implementer. The leaky bucket is a common abstraction to envision how policing is enforced. It consists of a bucket that holds a certain capacity of tokens. The maximum number of tokens that can be held by the bucket is generally referred to as b. The bucket is filled with tokens at a specified rate, called r. As a packet arrives at the associated queue, the bucket is checked to determine if there is an available token. If the bucket contains a token, it is removed and the packet is allowed to be transmitted. If the bucket is empty, the packet must wait until another token has been put into the bucket. The maximum number of packets that can be processed at a near simultaneous time is limited by the number of tokens that the bucket can hold. Since the size of the bucket is b, this provides a limit to the burst size being policed. For any time, t, the average rate is limited by the rate at which tokens are produced times t plus the number of tokens that are already in the bucket. A depiction of the basic leaky bucket structure is shown in Figure 9.

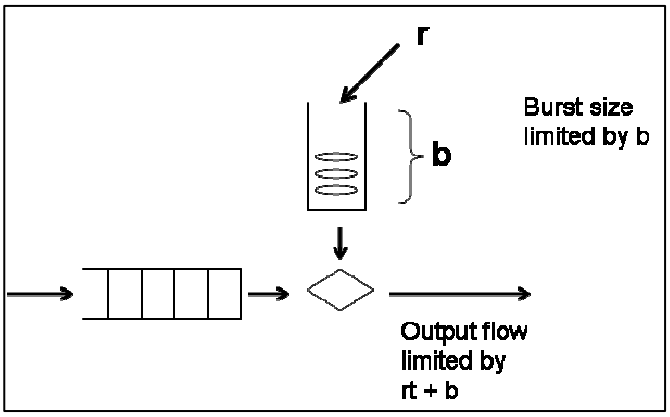

Figure 9. A Leaky Bucket Policy Implementer.

The basic leaky bucket structure provides controls for the average rate and the burst rate, but it does not necessarily police the peak rate. This control, if desired, can actually be handled quite simply. By implementing two leaky buckets in series, the peak rate may be controlled. The first leaky bucket can be implemented exactly as described above. This will then feed to

the second leaky bucket that controls the peak rate. A simple means of doing this is to have this second structure specifically designed so that the bucket only holds one token. It is then filled at a rate, p, which is equal to the peak rate desired. This second bucket will only allow one packet to be transmitted at a time while any other packet must wait for another token to be generated. Because the tokens are produced at the rate of the desired control level, the transmission of packets will never exceed the peak rate. The structure of this type of peak rate policing mechanism is shown in Figure 10.

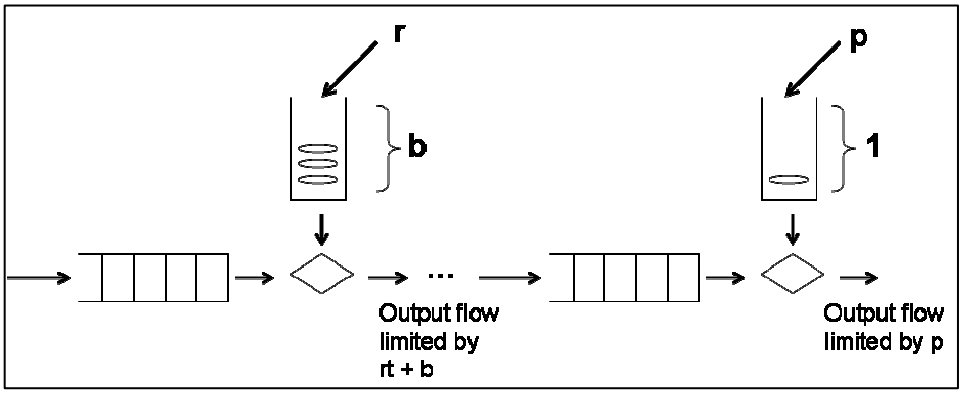

Figure 10. Peak Rate Policing.

2. Legacy Best Effort Service

The current Internet model provides traffic a best effort approach [8]. As packets arrive at a network router, they are placed in a FIFO queue for servicing. Packets are forwarded in the order in which they arrived without any special treatment given to time-sensitive flow requirements. Does this mean that the Internet is not capable of handling time-sensitive traffic? Obviously the answer is no because there are many applications that stream video, music, or voice traffic reasonably well. The primary reason this is possible is an overall lack of congestion. Most Internet links are specifically engineered to provide sufficient bandwidth so that overall usage does not exceed a certain percentage on average. There are still times, however, when congestion is experienced on the

Internet. Certain techniques have been adopted to allow time sensitive traffic to operate on best effort networks. Some of these techniques have been used extensively by military units operating over severely limited bandwidth tactical networks. A description of two common approaches to handling time sensitive traffic on best effort networks is provided to demonstrate some current practices that do not employ the QoS principles.

a. Application Level Controls

One of the most common approaches to handling time sensitive traffic on the Internet is by providing application level controls. This approach generally adjusts the performance of the networked application to account for delays and variance experienced in transiting the network. Application level controls do not provide a QoS solution because they do not provide any improved network service for the desired traffic flows. Instead, these solutions tend to minimize the impact of any network delays on the end user.

A very common application level control approach is to buffer the traffic so that it will appear to flow more fluidly at the receiving end. When the network experiences more congestion or delay, the buffer can be configured to increase in size accordingly. This can work very well for video or audio files that are being streamed from a server to a distant receiver. The additional delay created by the buffering is not significant because the flow is generally a one-way stream. This solution does not work as well for two-way voice or video teleconferencing. In these environments, the additional delay is very noticeable to the participants. Awkward pauses and overlapping conversations are commonly experienced as the buffering distorts the normal flow of human interaction. Therefore, while the use of buffers can be beneficial in certain situations, it certainly does not seem to be the best solution for two way audio and/or video exchanges.

Another application level control implementation involves the graceful degradation of audio or video. Faced with bandwidth constraints or

network congestion, this approach allows a flow to adjust the quality of its content. Generally, the intent is to reduce the sampling rate of the data stream rather than have the entire flow fail due to network problems. Provided that the reduction in quality is not too severe, the information being transmitted will still be useful to the end user. In the author's experience designing and managing tactical military networks, this approach has been used frequently for Unmanned Aerial Vehicle (UAV) video feeds. When the network cannot accommodate the bandwidth requirement of the video stream, the quality is reduced so that it requires less bandwidth. A usability problem arises when the video quality is reduced to the point that it is actually unusable at the distant end. While the video might seem acceptable on the screen of a laptop, when displayed on a screen at a Combat Operations Center it appears distorted and grainy. Additionally, because the degradation method actually lowers the quality of service, it is not suitable for applications where quality or data integrity are important.

b. Circuit Switching Approach

Another approach to providing service guarantees on best effort networks is to use circuit switching. This is typically done by reserving a portion of the transmission bandwidth solely for the use of the individual application. In military networks the bandwidth is generally programmed into the nodal multiplexers before pushing the signal onto the transmission equipment. A common application that might receive this treatment is the VTC between major commanders. The required bandwidth is set aside only for use by the VTC, thereby ensuring that it will not encounter congestion on its path. This approach generally works well for the supported application, but does so to the detriment of the rest of the network. It clearly violates the QoS principle to make efficient use of network resources since that portion of the bandwidth generally remains reserved when the application is not active. Another problem with this approach is that it generally relies on specific transmission paths for associated links and is

unable to dynamically route when a particular link goes down. A better approach would provide service guarantees to the desired application while still taking the basic QoS principles into account.

3. Integrated Services

One approach toward providing QoS guarantees is the Integrated Services (IntServ) model. IntServ is capable of providing multiple QoS guarantees within its structure. It provides individual flows quantitative guarantees on the QoS that will be provided. Within IntServ, there are three major classes of service: guaranteed service, controlled-load service, and best effort service. Best effort service operates as has already been described and will not be discussed further here. The other service classes, however, are worth examination to determine if they can be applied to the JMNO solution.

a. Guaranteed Service

The guaranteed service class used by IntServ "provides firm (mathematically provable) bounds on end-to-end datagram queuing delays [9]." To accomplish this task, guaranteed service must perform a call setup for each flow. It first characterizes the traffic that will be sent through the use of the leaky bucket mechanism previously discussed. This allows network nodes along the intended path to know the specification of the traffic that will be sent. It also provides a specification regarding the QoS it is requesting with a transmission rate at which packets will be sent. The service then reserves resources along the path where each router determines whether or not it is able to accommodate the request. Generally, this process is done through the resource ReSerVation Protocol (RSVP). Nevertheless, [9] does not specify the method used for the reservation and allows for other protocols or static configurations to be employed. If all routers along the path are able to provide the requested service, the flow will receive a guarantee on the maximum queuing delay it will experience. Each distinct flow would be handled by different queues at each network node.

b. Controlled-Load Network Service

Unlike guaranteed service, controlled-load network service provides predictive QoS. It seeks to provide a given flow with "a quality of service closely approximating the QoS that same flow would receive from an unloaded network element [10]." It uses an admission control process to only accept flows when the network is capable of providing it the desired level of service. Similar to guaranteed service, a controlled-load service flow provides a traffic characterization with a leaky bucket mechanism to inform downstream nodes of the traffic it can be expected to generate. It does not require separate queues for each flow, however, relying instead on managing the overall network load to allow traffic to be processed in a timely manner. Controlled-load network service does not provide any quantitative guarantees regarding QoS.

c. IntServ Assessment

While IntServ is able to provide QoS guarantees to supported flows, it is very complex to implement. It requires resources to be reserved at each node along the path to be effective. Even the specification of this technique admits that "guaranteed service is typically only useful if provided by every network element along the path [9]." Having all network routers implement IntServ and maintain state regarding all supported flows seems unadvisable for a military network. Additionally, IntServ does not provide the ability to define additional classes of traffic. While two flows might both desire guaranteed service, there is no mechanism for giving priority to the one that is more operationally critical. The IntServ solution appears to lack the specificity, scalability, flexibility, and simplicity that are desired for JMNO.

4. Differentiated Services

Differentiated Services (DiffServ) provides a compromise between best effort service and IntServ. DiffServ seeks a more flexible and scalable approach

to QoS by moving the more complex procedures (such as shaping, marking, and dropping packets, where necessary) away from core routers and toward edge routers [11], [12], [13]. Core routers are able to focus their efforts on forwarding packets based on markings created at the network edges. The following extracts from RFC 2475 provide the architectural requirements for DiffServ:

- Should accommodate a wide variety of services and provisioning policies, extending end-to-end or within a particular (set of) network(s),
- Should allow decoupling of the service from the particular application in use,
- Should work with existing applications without the need for application programming interface changes or host software modifications (assuming suitable deployment of classifiers, markers, and other traffic conditioning functions),
- Should decouple traffic conditioning and service provisioning functions from forwarding behaviors implemented within the core network nodes,
- Should not depend on hop-by-hop application signaling,
- Should require only a small set of forwarding behaviors whose implementation complexity does not dominate the cost of a network device, and which will not introduce bottlenecks for future high-speed system implementations,
- Should avoid per-microflow or per-customer state within core network nodes,
- Should utilize only aggregated classification state within the network core,
- Should permit simple packet classification implementations in core network nodes,
- Should permit reasonable interoperability with non-DiffServ-compliant network nodes,
- Should accommodate incremental deployment [14].

DiffServ provides a small number of classes to provide the desired network handling. The classes can be defined based on a flexible range of factors, including traffic type, source/destination IP address, and compliance with network specifications. [15] provides an in-depth analysis of DiffServ classes,

including recommended approaches for defining classification categories and implementing the classes into a network QoS solution. The following sections provide an overview of two primary DiffServ concepts: traffic classification and per-hop behavior.

a. DiffServ Traffic Classification

Packets entering a DiffServ network are marked based on the class of service it should receive. This marking is placed in the Type of Service (ToS) field in IPv4 and the Traffic Class field in IPv6. Six bits of the field are used to place a Differentiated Service Code Point (DSCP) that is used to determine the packet's class. The remaining two bits of the ToS or Traffic Class field are currently unused. DiffServ provides three categories of service classes that can be employed. These are Expedited Forwarding, Assured Forwarding, and Default Forwarding. Each category can be associated with different types of traffic and can provide varying levels of service.

(1) Expedited Forwarding (EF). Certain real-time traffic flows, such as voice or VTC streams, are inelastic regarding jitter and delay that might be encountered in a network. EF provides similar performance to the IntServ Guaranteed Service class. It allows associated flows to receive a logical link with a minimum guaranteed bandwidth value. Because EF receives its bandwidth guarantee independent of any other traffic, it is important to minimize the use of this category for the health of the overall network. The structure of DiffServ allows for more than one class to be assigned to the EF category. The general approach, however, is to define only one EF class [15]. This helps limit the use of EF to only those flows that actually need this service without abusing the network's health. Cisco routers have exactly one EF class defined with a standard DSCP value of 46 (binary 101110) [7].

(2) Assured Forwarding (AF). AF provides for the definition of multiple classes. Within each class, further subdivisions are possible to define drop preferences when congestion is encountered. Similar to what was seen

with EF, the DiffServ architecture allows for any number of AF classes. The standard approach is to define four AF classes [15]. Cisco follows this approach by establishing four classes with three drop preferences within each. The Cisco-defined classes with their drop preferences and DSCP values are shown in Table 10.

Class	Drop Preference	Identifier	DSCP (decimal)	DSCP (binary)
1	Low	AF11	10	001010
	Medium	AF12	12	001100
	High	AF13	14	001110
2	Low	AF21	18	010010
	Medium	AF22	20	010100
	High	AF23	22	010110
3	Low	AF31	26	011010
	Medium	AF32	28	011100
	High	AF33	30	011110
4	Low	AF41	34	100010
	Medium	AF42	36	100100
	High	AF43	38	100110

Table 10. Cisco DiffServ AF Classes.

(3) Default Forwarding (DF). The DF class is used to classify any packets that do not belong to the EF or AF classes. This traffic is generally provided best effort service, although it may be subject to traffic shaping or dropping if it does not comply with network specifications. There is only one DF class and it is assigned a DSCP value of zero (binary 000000).

With an understanding of the different DiffServ classes, it is possible to examine how packets are actually classified and assigned DSCP values. The classification process can be accomplished through a variety of methods and what is discussed here is just one approach. Other methodologies are discussed in [15]. The first step in this process is to determine which classes are desired to be used and what characteristics of the traffic should be used for assignment. For this example, we can assume that there is a need for EF service for the desktop VTC application, CU-SeeMe, which uses UDP packets on ports 7648 and 7649. We also might want to provide AF service to Internet Relay Chat (IRC) messages, which uses UDP packets on port 194. However, it might be desirable for IRC traffic from the local network to receive preference over those messages being generated by mobile units. We might, in this case, assign two AF classes to differentiate between these types of traffic. All other traffic will be in the DF class.

The next step for the classification process is to define access lists based on the characteristics that have been decided upon. There are multiple ways of defining access lists that vary based on desired effect and the version of the Internetwork Operating System (IOS) that is being used. The full breadth of access list definition is outside the scope of this thesis. The methods discussed here provide one example of one approach that might be useful to a JMNO implementation. Defining access lists is done through the global configuration mode with the format shown in Table 11.

Command	Purpose
Router(config)# access-list {*access-list-number*} {deny | permit} protocol <*source*> [<*source-mask*>] <*destination*> [<*destination-mask*>] [*operator destination-port*]	Example format for creating a generic access list
Router(config)# access-list 101 permit udp any any range 7648 7649	Creates access list 101, which allows UDP traffic from any address to any address with a destination port number in the range of 7648 – 7649
Router(config)# access-list 102 permit udp 192.168.32.0 0.0.31.255 any eq 194	Creates access list 102, which allows UDP traffic from network 192.168.32.0/19 to any address with a destination port number equal to 194 (Note: network mask is in one's complement form)
Router(config)# access-list 103 permit udp any any eq 194	Creates access list 103, which allows UDP traffic from any address to any address with a destination port number equal to 194 (Note: because rules are matched in order this does not conflict with access-list 102)
Router(config)# access-list 104 permit ip any any	Creates access list 104, which allows all IP traffic that has not been matched by the previous rules; Unless otherwise specified, all unmatched traffic receives best-effort service – defining a separate access list allows for shaping and handling of this traffic based on QoS policies

Table 11. Access List Configuration.

After the access lists have been created, it is necessary to map them to classes that can be used by DiffServ. It is sensible to follow the Cisco-defined classes because this allows logical organization that will provide more intuitive use when the classes are applied later. Creating the class maps begins in the global configuration mode. Each class could be mapped to multiple

access lists, although in this example they are only assigned one each. Similar to creating access lists, this is only one of many methods for accomplishing this task. The format for creating class maps is shown in Table 12.

Command	Purpose
Router(config)# class-map {match-all | match-any | *class-map-name*} <*class-map-name*> Router(config-cmap)# match access-group <*access-group number*>	Example format for creating a generic class map
Router(config)# class-map match-all EF Router(config-cmap)# match access-group 101	Associates access list 101 with the EF class
Router(config)# class-map match-all AF11 Router(config-cmap)# match access-group 102	Associates access list 102 with the AF11 class
Router(config)# class-map match-all AF21 Router(config-cmap)# match access-group 103	Associates access list 103 with the AF21 class
Router(config)# class-map match-all BE Router(config-cmap)# match access-group 104	Associates access list 104 with the BE class

Table 12. Class Map Configuration.

The fourth step in the traffic classification process is to associate the defined classes with a policy to mark the packets appropriately. This is done through the use of a policy map. A policy map allows specific actions or behaviors to be associated with the class. In this case, it is desired that the packet be marked with the appropriate DSCP value. Starting in the global configuration mode, the procedure to create a policy map is shown in

Table 13. The use of standard class and DSCP values helps avoid confusion in this process because Cisco routers automatically associate the DSCP number with the standard class name. For example, even when the value for the EF class is entered as shown below, if the configuration file is examined the entry will automatically display as "set ip dscp ef." Another note regarding this configuration concerns the assignment of DSCP value zero to the BE class. This automatically assigns this traffic to the DF class and would show up in the configuration file as "set ip dscp default." However, by creating this separate class it will be possible to perform policing actions at a later time, if desired.

Command	Purpose
Router(config)# policy-map <*policy-map-name*> Router(config-pmap)# class <*class-name*> Router(config-pmap-c)# set ip dscp <*dscp-value*>	Example format for creating a generic policy map
Router(config)# policy-map SETDSCP Router(config-pmap)# class EF Router(config-pmap-c)# set ip dscp 46	Creates a policy map named SETDSCP; Packets in class EF are marked with a DSCP value of 46
Router(config-pmap)# class AF11 Router(config-pmap-c)# set ip dscp 10	Packets in class AF11 are marked with a DSCP value of 10
Router(config-pmap)# class AF21 Router(config-pmap-c)# set ip dscp 18	Packets in class AF11 are marked with a DSCP value of 18
Router(config-pmap)# class BE Router(config-pmap-c)# set ip dscp 0	Packets in class BE are marked with a DSCP value of 0

Table 13. Policy Map Configuration.

The final step in traffic classification is to implement the policy on a specific interface. It is generally desired to apply the policy at the network's edge so that the packets are marked at the first router encountered in the DiffServ network. This is accomplished by entering the interface

configuration mode for the desired interface. The format for applying the packet marking policy at the network edge interface is shown in Table 14.

Command	Purpose
Router(config)# interface <*interface-name*> Router(config-if)# service-policy input <*policy-name*>	Example format for applying an input policy to a specific interface
Router(config)# interface e1/2 Router(config-if)# service-policy input SETDSCP	Applies the SETDSCP policy to packets arriving through interface Ethernet1/2

Table 14. Packet Marking at Network Edge.

b. DiffServ Per-Hop Behaviors

Defining classes and marking packets is useless unless there is a mechanism to use that information. DiffServ uses class definitions and packet markings to determine the Per-Hop Behavior (PHB) at each node. PHB defines a desired network forwarding behavior based on the packet classification. This can be implemented in the form of bandwidth guarantees or the use of particular scheduling structures. DiffServ does not dictate which methods are employed to provide the forwarding behavior [15]. A common approach is to use WFQ to provide each class a portion of the available bandwidth. One queue can be allocated for EF traffic and four additional queues can be used by the AF classes. One final queue is reserved for all DF traffic. The EF queue provides a minimum bandwidth rate. The AF queues each can be assigned a percentage of the available bandwidth. Within each AF class, each division of the class is assigned a drop preference with the highest values being dropped first when there is congestion. The DF queue receives best effort service after all other classes have received their service guarantees.

Implementing the desired PHBs begins by mapping the assigned DSCP values to appropriate classes. These classes will be defined to match the desired queues that will be created. A common approach is to name these classes after precious metals in a manner that clearly indicates the relative value of each. Platinum is the highest priority class and can be mapped to handle all EF traffic. AF traffic can be handled by the gold, silver, and bronze classes. In the author's experience, generally, only three AF queues are created because there is typically not a need for more than this and there is a desire to keep the structure as simple as possible. If needed, a fourth class, say tin, could be created as well. The final class will be simply named best-effort to handle all DF traffic. The class map is created through the global configuration mode by creating the class and assigning traffic to it based on the DSCP value that was encoded on each packet previously. The format for creating the PHB classes is provided in Table 15.

Command	Purpose
Router(config)# class-map match-all <*class-name*> Router(config-cmap)# match ip dscp <*dscp-value(s)*>	Sample format for creating PHB classes based on DSCP values
Router(config)# class-map match-all platinum Router(config-cmap)# match ip dscp 46	Creates platinum class map for packets with a DSCP value of 46
Router(config)# class-map match-all gold Router(config-cmap)# match ip dscp 10 12 14	Creates gold class map for packets with a DSCP value of 10 12 14
Router(config)# class-map match-all silver Router(config-cmap)# match ip dscp 18 20 22	Creates platinum class map for packets with a DSCP value of 18 20 22
Router(config)# class-map match-all bronze Router(config-cmap)# match ip dscp 26 28 30	Creates platinum class map for packets with a DSCP value of 26 28 30
Router(config)# class-map match-all best-effort Router(config-cmap)# match ip dscp 0	Creates platinum class map for packets with a DSCP value of 0

Table 15. PHB Class Definition by DSCP Value.

After the PHB classes have been established, they should be mapped to the specific forwarding behavior that is desired. For example, the VTC that is receiving EF service might require a guaranteed minimum bandwidth of 256 KBps (Kilobytes per second). The platinum class would then be mapped to a policy that provides this service. The gold, silver, and bronze AF classes might want to receive 35, 20, and 20 percent of the available bandwidth respectively. It may also be desired that within each AF class that the queues employ Weighted Random Early Detection (WRED). This allows the queue to utilize the drop preference assigned to packets when congestion occurs. Cisco routers use tail drop unless otherwise specified, thereby causing the most recent arrival packets to be dropped when the queue is full. It is also possible to conduct traffic shaping and policing with the policy map, if desired. The bronze class, for example, could be shaped to an average rate of 100 KBps. Policing of DF traffic in the best-effort class can establish an average rate, normal and maximum burst rates, and actions to take when the class conforms, exceeds, or violates the policy. The configuration for creating a PHB policy map is shown in Table 16.

Command	Purpose
Router(config)# policy-map JMNO	Creates a policy map named JMNO
Router(config-pmap)# class platinum Router(config-pmap-c)# priority 256	Assigns the platinum class a bandwidth guarantee of 256 Kbps
Router(config-pmap)# class gold Router(config-pmap-c)# bandwidth percent 35 Router(config-pmap-c)# random-detect dscp-based	Assigns the gold class an allocation of 35% of available bandwidth; Implements WFQ based on DSCP values
Router(config-pmap)# class silver Router(config-pmap-c)# bandwidth percent 20 Router(config-pmap-c)# random-detect dscp-based	Assigns the silver class an allocation of 20% of available bandwidth; Implements WFQ based on DSCP values
Router(config-pmap)# class bronze Router(config-pmap-c)# bandwidth percent 20 Router(config-pmap-c)# random-detect dscp-based Router(config-pmap-c)# shape average 100000	Assigns the bronze class an allocation of 20% of available bandwidth; Implements WFQ based on DSCP values; Shapes traffic to an average of 100 Kbps
Router(config-pmap)# class best-effort Router(config-pmap-c)# police 56000 1750 1750 conform-action transmit exceed-action drop violate-action drop	Polices best-effort packets to an average rate of 56 Kbps, a normal and maximum burst size of 1750 Bytes, transmits packets when they conform with policies and drop packets when they exceed or violate the policies

Table 16. PHB Policy Map Definition.

Finally, the PHB policy must be assigned to an interface where it will be enacted. Generally, this will be done on a core router's outbound interface. The procedure is very similar to the way the inbound policy is mapped to an interface for packet classification. The format is shown in Table 17.

Command	Purpose
Router(config)# interface <*interface-name*> Router(config-if)# service-policy output <*policy-name*>	Example format for applying an output policy to a specific interface
Router(config)# interface Ethernet0/0 Router(config-if)# service-policy output JMNO	Applies the JMNO policy to packets departing through interface Ethernet0/0

Table 17. DiffServ PHB Assignment to Outbound Interface.

This chapter provides some details about network routing and QoS solutions that might be used by JMNO. It describes the use of protocols, such as EIGRP, BGP, and DHCP, that can be used for networking solutions. It also examines the principles of QoS and looks at best effort, IntServ, and DiffServ solutions. Based on the investigation described in this chapter, certain approaches seem to be preferable over others. Within the networking solutions, the use of BGP and DHCP appear to be of value for solving specific JMNO problems. While EIGRP is a valuable tool, this thesis seeks to use it only for internal routing and does not implement it for lateral link or mobile unit connections. Likewise, the QoS solutions provide different approaches where each has its own strong points. The preferred approach for this thesis, however, is DiffServ. Chapter III and Chapter IV provide specific solutions that are implemented in the laboratory network for network routing and QoS designs respectively.

THIS PAGE INTENTIONALLY LEFT BLANK

III. PROPOSED NETWORK ROUTING CONFIGURATION

Before any QoS solution can be implemented, there must be an existing network on which it will operate. In working toward the QoS objectives during this research, it became apparent that there was a strong need for routing standards to be established. This would allow the backbone network to function as required for JMNO functionality. In order to provide seamless connectivity for both inter-Service lateral links and mobile users transiting the battlespace, there should be an agreed upon set of standards. This will minimize the need for ad-hoc network solutions to be negotiated between tactical units. Instead, network operators should be able to refer to the joint TTP to find the solution that best fits their operating environment. This chapter defines some potential routing configuration solutions that could be adopted for JMNO standards.

The network structure that provides the basis for this research is based on a JMNO architecture used for the Joint User Interoperability Communications Exercise, July 2007. The architecture has the four military Services connected through a JTF node, which then provides access through STEP sites to the DISN network. It also implements lateral links between Services at the tier seven layer. The laboratory network that is used does not simulate all of the Service nodes. This is because of equipment availability constraints and since the primary goal is to demonstrate the viability of the proposed solution – not to implement the entire structure. As a result, only the Army and Marine Corps networks are included along with connectivity through the JTF to the DISN network. The entire network is isolated, with no real-world connectivity. The Army units included are the Army Forces (ARFOR) headquarters, an Army Brigade (ARMY_BGDE), and two Army Battalions (ARMY_BN1 and ARMY_BN2). The Marine Corps units included in the lab network are the Marine Forces (MARFOR) headquarters, a Marine Corps Regiment (MAR_REGT), and two Marine Corps Battalions (MAR_BN1 and MAR_BN2). For this network, the battalions of each Service are

considered the mobile units. The network nodes that are included and their associated connectivity links are depicted in Figure 11.

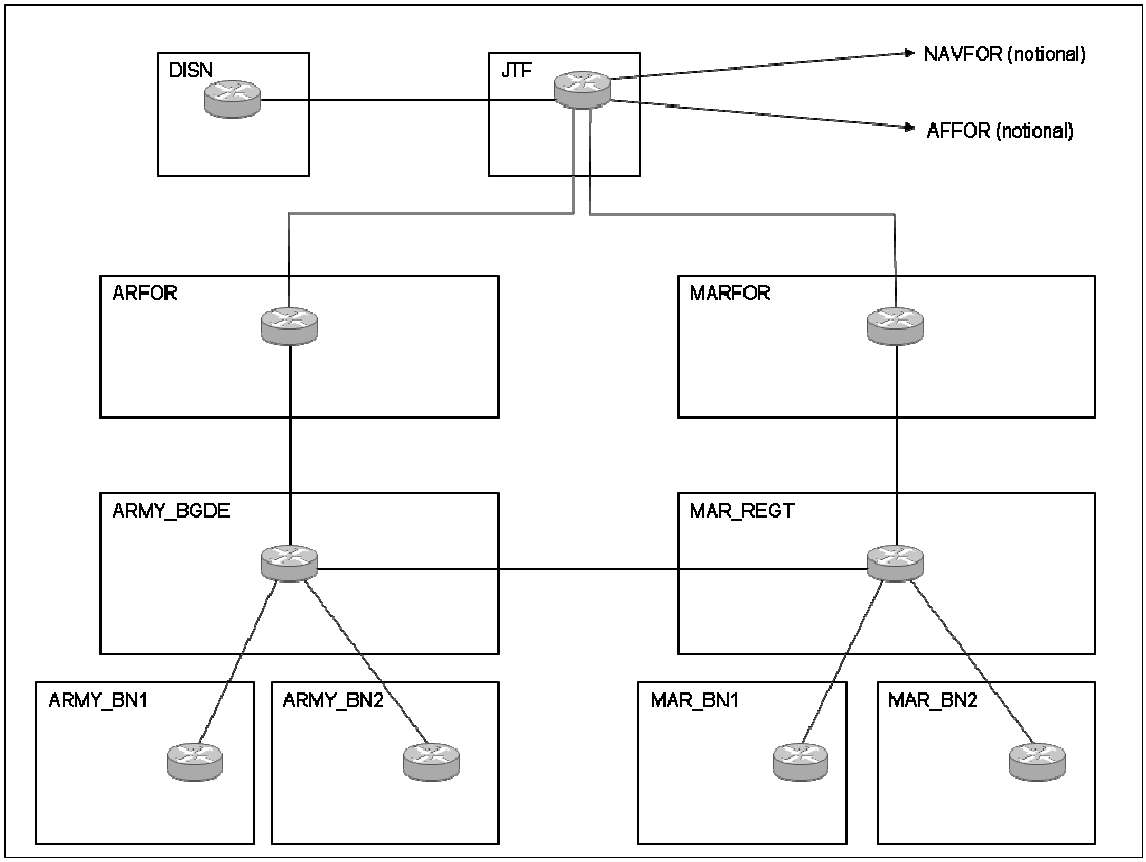

Figure 11. Initial Laboratory Network Connections.

A. BORDER GATEWAY PROTOCOL FOR JMNO ROUTING

Providing a routing solution between autonomous systems in JMNO requires a protocol that is scalable and able to interact with diverse interior routing protocols. For example, the Air Force might be running EIGRP on its internal networks while the Navy is running OSPF. As discussed in Chapter II, BGP is well suited to provide the connectivity between these different networks. BGP does this effectively throughout the Internet and is currently used for higher level military network connections, as well. Additionally, BGP is an open standard protocol, in contrast to EIGRP which is a Cisco proprietary solution.

This solution proposes to adopt BGP for lower level units in order to enable lateral inter-Service connections as well as internal connections that provide flexibility for mobile units.

1. External Border Gateway Protocol Configuration

To begin constructing the BGP structure, it is necessary to understand where the AS boundaries should reside. Because JMNO desires that tiers seven and eight be separated from their higher command units, it makes sense to place them within their own AS. This allows the use of External Border Gateway Protocol (eBGP) to provide connections to higher and lateral units. The approach taken for the laboratory network is depicted in Figure 12. For simplicity, DISN is depicted as a single node within a single AS. This should be considered as the first node and AS encountered within the larger DISN structure, rather than as a representation of the entire DISN network.

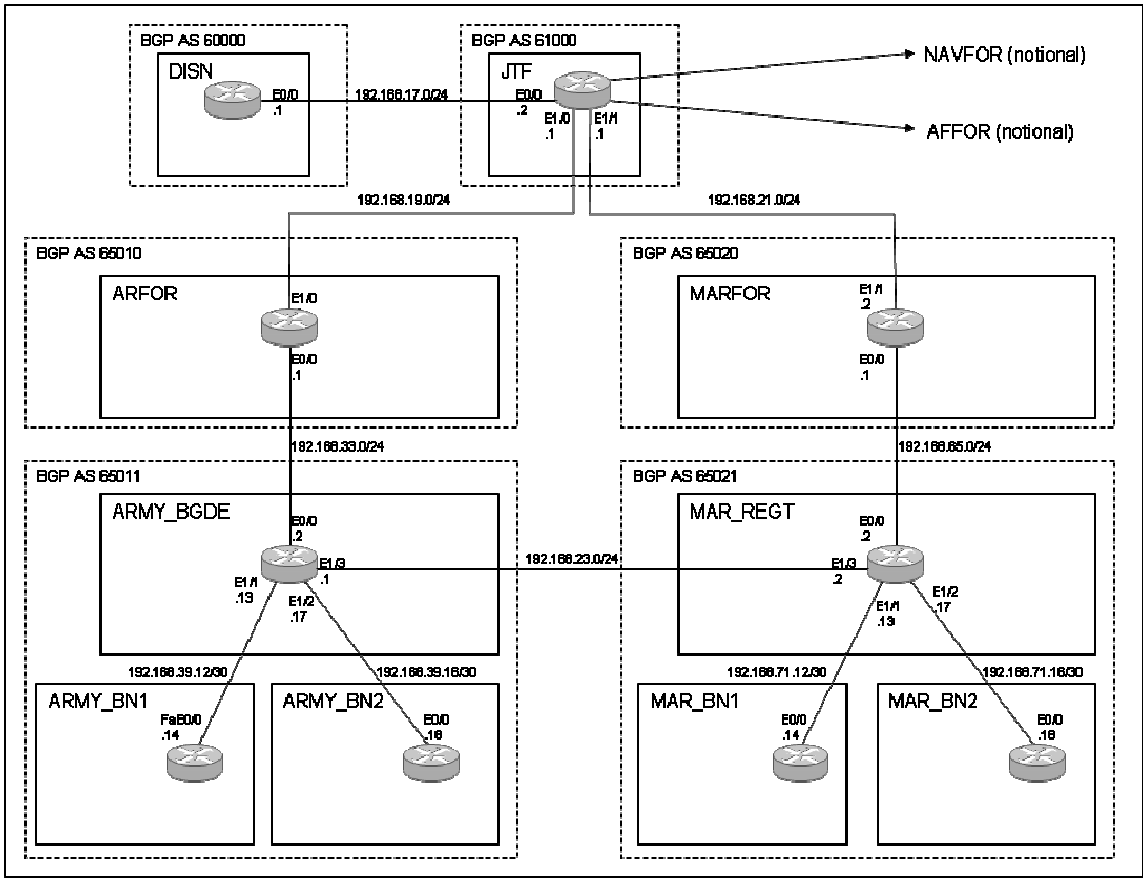

Figure 12. Sample BGP Autonomous Systems.

Only those nodes that have a link connecting to a different AS are required to implement eBGP. In this network, the eBGP participants are DISN, JTF, ARFOR, ARMY_BGDE, MARFOR, and MAR_REGT. The AS numbers for DISN and JTF are taken outside of the private range because those units are assumed to have official, registered numbers assigned. The other units are assigned AS numbers based on the JMNO allocations discussed in Chapter I.

The implementation of eBGP is relatively straightforward, although many optional configurations are possible. For the JMNO solution, the goal is to keep the method as simple as possible. The approach here, accordingly, is to begin with the least complex design possible and to build up the complexity only where it is required. As complexity increases, the implementation could be accomplished through scripts, dynamic executable programs that generate the

IOS commands and send them to the appropriate router based on inputs from the operator/administrator. This capability is left as an area for further study. The initial eBGP configurations for the Army nodes are shown in Table 18. The configuration of the corresponding Marine Corps units follows the same format as the ones depicted.

Router	Configuration
ARFOR	router bgp 65010 network 192.168.32.0 neighbor 192.168.19.1 remote-as 61000 neighbor 192.168.19.1 soft-reconfiguration inbound neighbor 192.168.33.2 remote-as 65011 neighbor 192.168.19.1 soft-reconfiguration inbound
ARMY_BGDE	router bgp 65011 network 192.168.34.0 neighbor 192.168.33.1 remote-as 65010 neighbor 192.168.19.1 soft-reconfiguration inbound neighbor 192.168.23.2 remote-as 65021 neighbor 192.168.23.2 soft-reconfiguration inbound

Table 18. Router Configurations for eBGP.

An additional requirement for JMNO is the ability to control the advertisement of lateral link routes. The tier seven peers, ARMY_BGDE and MAR_REGT, do not want to advertise routes they have learned from one another to their higher headquarters. As discussed in Chapter II, BGP provides a mechanism for filtering route advertisements based on the AS path attribute. To implement a path filter, it is necessary to define the parameters of the desired filter and then to apply it to the particular eBGP neighbor. The configuration settings used on MAR_REGT are shown in Table 19. Corresponding settings are also applied to the ARMY_BGDE router.

Router Command	Purpose
MAR_REGT(config)# ip as-path access-list 1 deny _65011_	Defines a BGP related access list which denies any route containing AS 65011
MAR_REGT(config)# ip as-path access-list 1 permit .*	Adds a rule to the previously defined list allowing all other routes
MAR_REGT(config)# router bgp 65021	Enter router configuration mode for MAR_REGT's BGP instance
MAR_REGT(config-router)# neighbor 192.168.65.1 filter-list 1 out	Establish a BGP path filter that applies the access list rules on outbound route advertisements

Table 19. Path Filter Configuration.

After the router implements the filter, it should reset its BGP configuration so that old routes are dropped. This is accomplished by using the "clear ip bgp *" command. To verify that the path filter successfully accomplished the desired effect, it is necessary to view the routing table at the higher level unit before and after the filter is put in place by using the "show ip route" command. Applicable portions of the ARFOR routing table are shown, before and after ARMY_BGDE implemented its filter, in Figure 13. Although not shown, ARMY_BGDE, ARMY_BN1, and ARMY_BN2 are all still able to use the lateral link to route to the Marine units. ARFOR, on the other hand, is forced to route traffic through JTF to reach the Marine units. It is still able to route to the other Army units through its link to ARMY_BGDE.

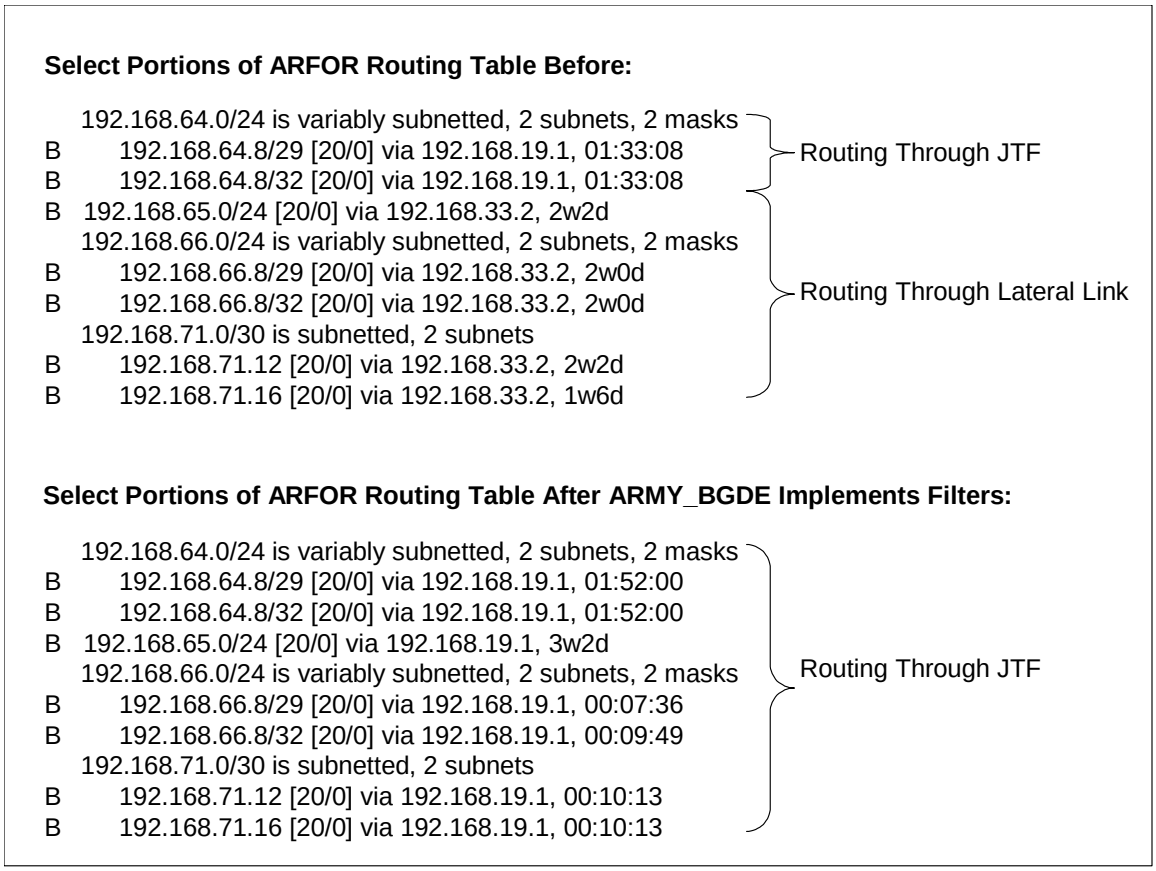

Figure 13. Effect of Path Filters on Routing Table Entries.

2. Internal Border Gateway Protocol Configuration

The tier seven and eight units for each Service have been combined into a common iBGP configuration. Within the iBGP AS, the nodes are joined by a common interior routing protocol. This interior protocol provides a means for iBGP peers to exchange information when they are not directly connected. In the laboratory configuration, the protocol used for this is EIGRP. Alternatively, any appropriate protocol, such as Open Shortest Path First (OSPF) or Routing Information Protocol (RIP), could be used instead. If desired, interior routing could also be implemented through the creation of static routes. The use of an interior routing method provides a means for iBGP peers to communicate with one another.

Within a particular AS, all iBGP peers must be fully meshed. Each must define neighbor relationships to all of the others. They also must be able to reach the others through the interior routing or be directly connected to all others. Because it is unlikely that all iBGP peers will have direct connections with one another (particularly since it is not scalable for large networks), the preferred method is to use interior routing. In configuring the peer relationships, it is recommended that each router use a loopback address for establishing the sessions. This permits the relationship to be maintained when external interfaces change states; otherwise it is possible for a router's identification to be inadvertently changed as, unless specifically specified as a loop-back address, it defaults to the highest operational IP address configured on the router.

The procedures for establishing iBGP neighbor relationships follow the same general format described for eBGP. In this case, however, the remote AS number is the same as the local AS number. The laboratory implementation for this thesis uses loopback addresses for iBGP relationships. The iBGP configurations for MAR_REGT, MAR_BN1, and MAR_BN2 are shown in Figure 14. The implementation for the corresponding Army units follows the same format.

In a larger network structure, the tier seven and eight units could potentially be divided further into additional AS structures. For example, the Marine Corps units might have separate ASs for the Marine Division, Marine Aircraft Wing, and Marine Logistics Group. These major subordinate commands could implement their own interior routing protocols and create neighbor relationships with one another. The procedures would follow the eBGP configurations described above. This approach would allow the most flexibility and reduce the number of iBGP relationships that would need to be defined. Other, equally valid, approaches could include the use of BGP confederations or route reflectors. These methods are not used in this thesis and, for brevity, are not discussed in detail.

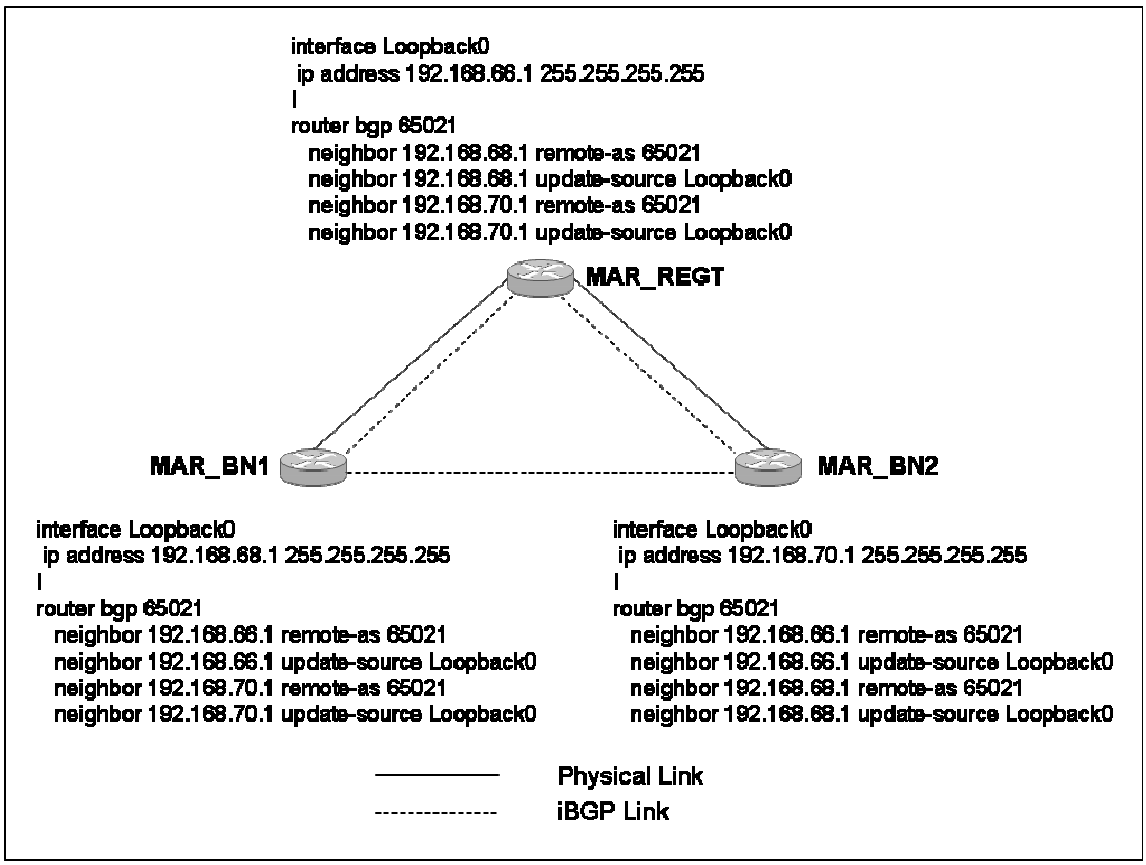

Figure 14. iBGP Configuration.

B. DYNAMIC HOST CONFIGURATION FOR MOBILE ROUTERS

With a core network solution in place, it is possible to look for methods to make it easier for mobile units to establish connectivity. One means of accomplishing this is by employing DHCP to assign IP addresses to the external interface that a mobile unit uses to connect to the hosting or supporting unit. By allowing the mobile unit to receive its interface IP through DHCP, it is able to move to a new location and draw a new address without changing its configuration. The unit's backside network addresses do not change upon moving to the new network. Only the connecting interface uses DHCP to assign its IP address. This allows the mobile unit to avoid any manual configuration changes when it moves to a new location. This is particularly important for JMNO units that might migrate connections between ASs.

1. Host Service Router Configuration

The router that provides the DHCP service to mobile units must be configured to issue IP addresses in a simple manner. It is also desirable for the implementation to efficiently use the address space in order to avoid wasting valuable IP addresses that tactical units often do not have in abundance. The proposed solution, therefore, provides firm address allocations to specific interfaces with minimal overhead costs. Each interface is assigned to a particular subnet with a 30-bit network prefix, allowing for two assignable addresses. The interface itself is statically assigned the first of the addresses. This leaves exactly one address that may be assigned to a mobile unit that connects to the interface. The precise address of a mobile unit's interface is known whenever it is connected to the host.

The configuration settings to enable DHCP service on ARMY_BGDE are shown in Table 20. Similar settings are configured on each interface that should assign IP addresses through DHCP. It is recommended that this procedure be conducted on each interface that might have a mobile unit connect to it. This configuration should also be used for host Service units that provide connections to units that might later move to another location. This allows the mobile unit to implement its configuration as a DHCP client before it moves to another tactical location. The underlying physical layer could be wireless – eliminating wired connection requirements.

The laboratory network for this thesis provides DHCP on all open interfaces on the ARFOR, ARMY_BGDE, MARFOR, and MAR_REGT routers. Additionally, the connection to each battalion is created through DHCP to allow these units to already be configured as DHCP clients in the event that they move to another location.

Command	Purpose
ARMY_BGDE(config)# ip dhcp pool ARMY_BGDE_E1/0	Establishes the DHCP pool with the desired name
ARMY_BGDE(dhcp-config)# network 192.168.39.12 255.255.255.252	Adds the network address space that DHCP is assigned
ARMY_BGDE(dhcp-config)# default router 192.168.39.13	Sets a specific interface address to issue IPs from this DHCP pool
ARMY_BGDE(config-router)# exit	Returns to the global configuration mode from the DHCP configuration mode
ARMY_BGDE(config)# ip dhcp excluded-address 192.168.39.13	Excludes a particular address from being assigned by DHCP, corresponding to the server's address
ARMY_BGDE(config)# interface e1/0	Enters the interface configuration mode for the desired interface
ARMY_BGDE(config-if)# ip address 192.168.39.13 255.255.255.252	Assigns the same IP address for the DHCP default router setting
ARMY_BGDE(config-if)# no shutdown	Ensures the interface is in active mode

Table 20. Host Router DHCP Settings.

2. Mobile Router Configuration

Establishing a mobile unit interface to accept its IP address from a DHCP server is very simple. The router is simply set to act as a DHCP client on the desired interface. This is accomplished from the global configuration mode by following the commands shown in Table 21. The example settings are established on MAR_BN1, which has been defined as a mobile unit router. A similar procedure is implemented on each individual mobile router. For the laboratory network each battalion is considered a mobile unit that could displace into another Service's area of operations.

Command	Purpose
MAR_BN1(config)# interface e0/0	Enters the interface configuration mode for the desired interface
MAR_BN1(config-if)# ip address dhcp client-id ethernet0/0	Configures the interface to accept its IP address from DHCP; Provides a client ID indicating the specific interface to the DHCP server
MAR_BN1(config-if)# no shutdown	Ensures the interface is in active mode

Table 21. Mobile Router Configuration Settings.

3. Procedures for Mobile Unit Connection

For a mobile unit to use the DHCP service, the server and client interfaces must be configured as described in the previous two Sections. When the mobile unit moves to a new location it will negotiate a new IP address with the new DHCP server process. If the router does not automatically seek a new address, this can be done manually by following the same procedures for creating the DHCP client relationship described in Section B.2 of this chapter. This causes the router to drop its previous address and to negotiate for a DHCP-assigned address with the available server.

At this point, the host router needs to know which network(s) the mobile unit needs to have advertised to the rest of the network. This is the only step for the mobile connection that requires manual changes by the network administrators. There are two options that may be followed for the advertisement to take place. The first is that the host router creates a static route to the mobile network through the connecting interface. It also must ensure that it includes a statement in its BGP configuration to redistribute static routes. The mobile router must have a static route included that pushes all outbound traffic through the interface. Another option is for each router to add eBGP neighbor statements to its configuration, creating a new eBGP peer relationship between the networks.

The procedure for building this relationship follows the example provided in Section A.1 of this chapter. An example of how this procedure could be implemented can be seen if MAR_BN1 were to move into the Army sector and connect to an available DHCP-enabled interface on the ARFOR router. It is possible to assume that the ARFOR interface was assigned IP address 192.168.39.1 and that DHCP is able to assign 192.168.39.2 to a connecting router. MAR_BN1 will automatically negotiate this assignment upon connection. Next, the network administrators are required to determine and configure the AS number of each network. In this case, ARFOR is assigned AS 65010 and MAR_BN1 belongs to AS 65021. Manual configuration would be made on each router as indicated in Table 22.

Router	Configuration
ARFOR	router bgp 65010 neighbor 192.168.39.2 remote-as 65021 neighbor 192.168.39.2 soft-reconfiguration inbound
MAR_BN1	router bgp 65021 neighbor 192.168.39.1 remote-as 65010 neighbor 192.168.39.1 soft-reconfiguration inbound

Table 22. eBGP Settings for Mobile Router Connections.

At this point, the network is fully configured with a working routing solution. Lateral links have been established between the Services with appropriate path filters to manage route advertisements as desired by JMNO. Appropriate eBGP and iBGP peering relationships have been established between neighbors. Also a DHCP solution for minimizing configuration changes on mobile unit routers has also been implemented. A complete diagram of the configured network is shown in Figure 15. It is now possible to explore QoS solutions based on the operational network.

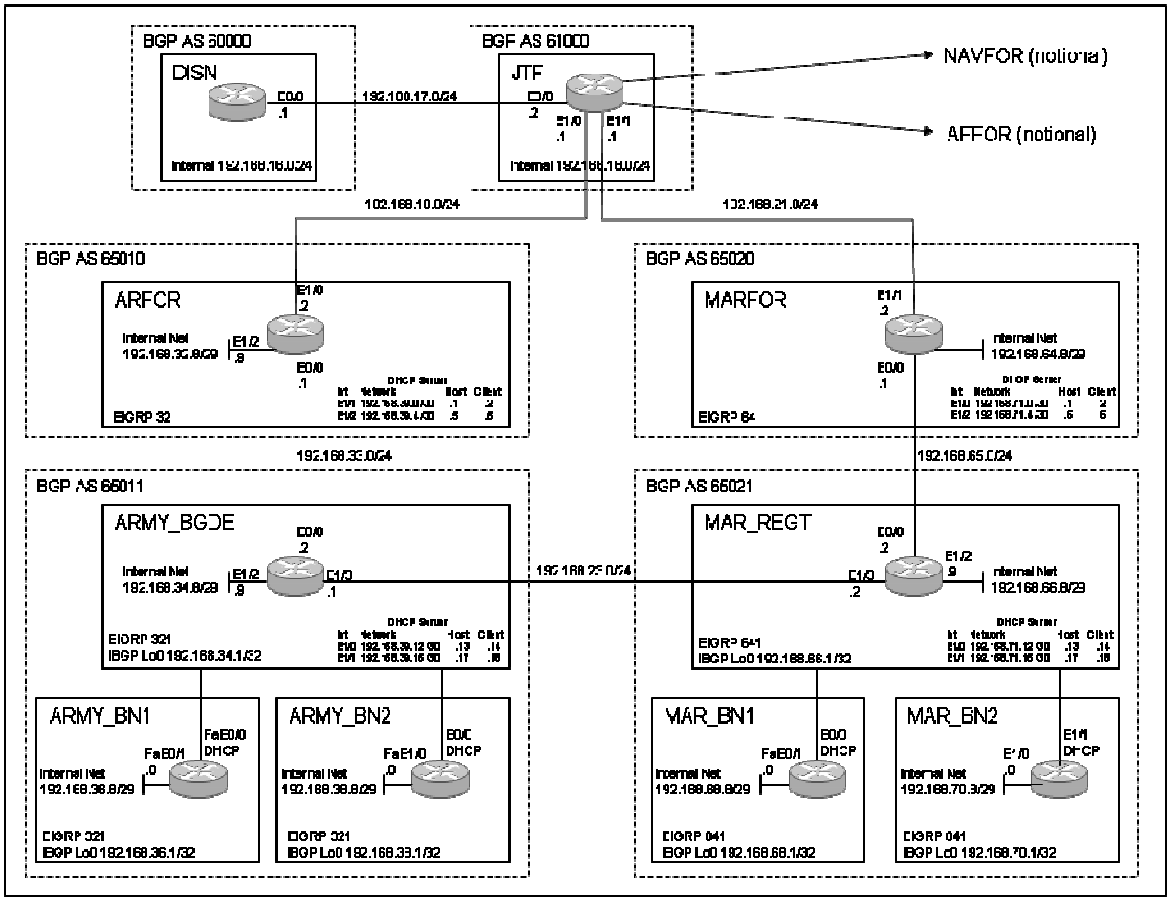

Figure 15. Final Network Routing Solution.

IV. PROPOSED QUALITY OF SERVICE METHOD

This thesis recommends that JMNO adopt the use of DiffServ to provide QoS. DiffServ provides service guarantees that are not available with best effort approaches without the complexity and scalability issues of IntServ. DiffServ provides a means of granting priority handling to critical time-sensitive flows across the network. It also allows flexibility for ensuring that the traffic from a host unit is not adversely affected by excessive mobile unit use.

The laboratory network described in Chapter III is used to implement DiffServ. This creates the need for a slight modification from a real-world network implementation. In practice, there would be multiple routers at a node such as ARMY_BGDE. This would allow packet marking to take place on routers that accept incoming traffic while other routers implement the desired PHBs for each class. It is recommended that this approach be used for JMNO because mobile units might classify traffic differently than the local network policies. The limited network structure in the laboratory, however, dictates an alternative approach. To separate the edge router actions from the core router actions, all incoming traffic is generated from networks connecting at the battalion level. This allows the higher level units to act solely as core routers and to avoid the packet classification process. The laboratory method allows the viability of the solution to be studied, even though in a real-world network the same actions might be taken at different locations.

A. DIFFSERV CLASS ORGANIZATION

The most important step in establishing DiffServ is defining a framework for the classes and behaviors that are desired. Decisions made at this point impact all of the configurations that provide the implementation. Changing the framework later can be extremely cumbersome and increases the likelihood of configuration errors. It is recommended, therefore, that a minimalist approach that will accomplish the desired goals be used. It is simpler to add additional

classes and behaviors than it is to change existing ones. The laboratory approach described here, on the other hand, is moderately robust and is designed to demonstrate the capabilities of DiffServ rather than act as a model for a full implementation. All of the protocols and bandwidth assignments in this chapter are intended purely as examples and should not be configured without conducting a detailed needs analysis on any operational network.

1. Traffic Classification

Network planners should coordinate with the operational users to determine which classes are needed. This requirements analysis assists in creating a DiffServ configuration that provides the appropriate service levels required by the units involved. This section provides a fictional scenario that demonstrates the analysis phase that supports the laboratory network for this thesis. For example, it might be decided that the most critical application on the network is the VTC used to coordinate between commanders. In this case, the example application used is CU-SeeMe, which operates through UDP on ports 7648, 7649, or 24032. The required bandwidth for this application is 256 Kbps. Without DiffServ support, the means to ensure support for the VTC by the units involved is through circuit switching, either physical or virtual.

The operators also indicate that Voice over IP (VoIP) and IRC are critical applications that should receive special handling. When conflict between these applications occurs, it is desired that the IRC traffic be dropped first. It is also decided that these applications should provide different service to local units as compared to mobile units. The VoIP application that is used sends UDP packets through a range of ports, from 16384 to 32768. The IRC application sends UDP packets through port 194. After analyzing average usage statistics, it is determined that the local units should receive 35 percent of the bandwidth while mobile units are allocated 20 percent for these applications.

Additionally, there is a need to provide a level of service to certain critical, but not time-sensitive, applications. These applications are Simple Mail Transfer

Protocol (SMTP), Post Office Protocol 3 (POP3), and Simple Network Management Protocol (SNMP). These applications all use TCP to send traffic. SMTP uses port 25 and POP3 uses port 110. SNMP uses ports 161 and 162. The statistical analysis indicates that these applications should be provided with 20 percent of the available bandwidth. It is determined within this category it is preferred that mobile unit traffic be dropped before traffic from local units.

The final determination of the requirements analysis phase is that best-effort traffic should be policed to conform to network policies. This traffic is to be limited to an average rate of 56 Kbps with normal and maximum burst rates limited to 1750 Bytes each. When the best-effort flows conform to these policies the packets will be sent. Otherwise the packets should be dropped.

Based on the analysis phase, the network planners can determine how to create a DiffServ structure that is able to support the requirements that have been established. These planners determine that the VTC application should be supported by EF with no differentiation between host network and mobile unit traffic. Host network VoIP and IRC traffic should be provided the highest AF handling with IRC packets being dropped before VoIP ones. The next highest class should be for VoIP and IRC traffic originating from mobile units and it should follow the same drop precedence used for host traffic. The next class contains SMTP, POP3, and SNMP traffic with mobile unit traffic having higher drop precedence than host network traffic. All other traffic falls under a best-effort class.

2. Forwarding Behavior

The analysis phase provides the building blocks to define the desired forwarding behavior. This can be translated into specific PHB rules for the different classes. The first step in establishing PHB policies is to define the class maps that are needed. A platinum class can be created for the VTC traffic. Host network VoIP and IRC traffic can be placed in a gold class. VoIP and IRC traffic

from non-local IP addresses can use a silver class. A bronze class can be used for SMTP, POP3, and SNMP traffic. All other IP traffic will be placed in the best-effort class.

The policy map will establish a PHB for each class defined above. The platinum class can be provided a static bandwidth guarantee to provide EF service. The gold class should implement AF service to provide it an appropriate bandwidth percentage. It is also desired that this class implement a drop preference between the types of traffic within the class. This can be handled by implementing WRED based on the DSCP value marked on the packet at the edge router. The silver and bronze classes are similar to the gold class in providing AF service with DSCP-based WRED. The best effort class will police traffic based on the parameters determined during the analysis phase.

B. EDGE ROUTER CONFIGURATION

The edge routers are responsible for classifying traffic entering the network. Correct marking at the network edge is important for flows to receive the appropriate handling by the DiffServ-enabled network nodes. This entails assigning traffic to access lists based on the desired characteristics, mapping those access lists to specific classes, mapping the classes to DSCP marking policies, and assigning the policies to the incoming interface.

1. Access List Assignment

The format for creating access lists is discussed in Chapter II. The access list translates the pertinent information from the analysis phase and places it into the router configuration for traffic characterization. Because this action is desired to take place at all edge routers, it will be conducted on ARMY_BN1, ARMY_BN2, MAR_BN1, and MAR_BN2. Each node will use the same configuration, adjusted for local IP addresses, so that packets are marked

consistently across the network. All entries for creating the access lists are done from the global configuration mode. The access lists for ARMY_BN1 are shown in Table 23.

Command	Purpose
access-list 101 permit udp any any range 7648 7649 access-list 101 permit udp any any eq 24032	Matches all CU-SeeMe traffic from any host
access-list 102 permit udp 192.168.32.0 0.0.31.255 any range 16384 32768	Matches VoIP traffic from local network addresses
access-list 103 permit udp 192.168.32.0 0.0.31.255 any eq 194	Matches IRC traffic from local network addresses
access-list 105 permit udp any any range 16384 32768	Matches VoIP traffic that was not matched previously
access-list 106 permit udp any any eq 194	Matches IRC traffic that was not matched previously
access-list 108 permit tcp 192.168.32.0 0.0.31.255 any eq 25 access-list 108 permit tcp 192.168.32.0 0.0.31.255 any eq 110 access-list 108 permit tcp 192.168.32.0 0.0.31.255 range 161 162	Matches all SMTP, POP3, and SNMP traffic from local addresses
access-list 109 permit tcp any any eq 25 access-list 109 permit tcp any any eq 110 access-list 109 permit tcp any range 161 162	Matches SMTP, POP3, and SNMP traffic that was no matched previously
access-list 111 permit ip any any	Matches all other IP traffic

Table 23. Access Lists for ARMY_BN1 Router.

2. Class-Based Packet Marking

The remaining procedures for configuring traffic classes are relatively straightforward. First, it is necessary to match access groups to classes. Second, the classes are mapped to policies for marking packets with the

appropriate DSCP value. Lastly, the desired interface is assigned to implement the policy. These procedures follow the format described in Chapter II. The laboratory network implements the configurations as depicted in Figure 16.

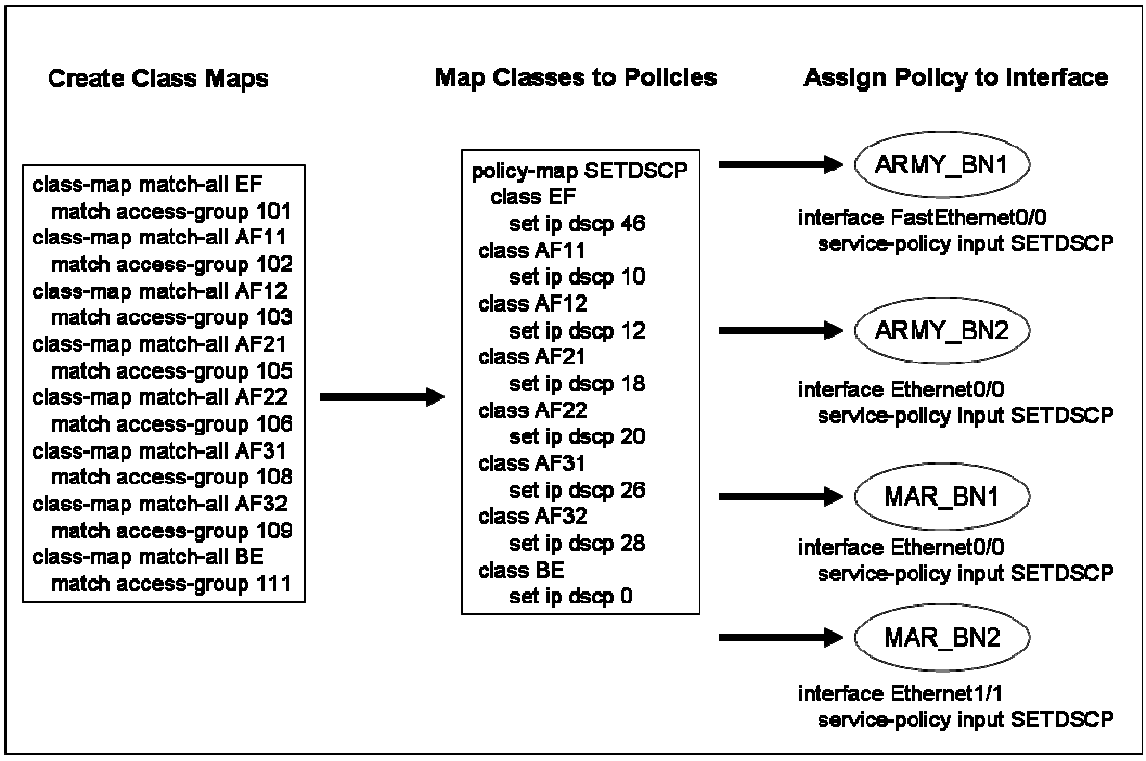

Figure 16. Class-Based Packet Marking Configuration for Edge Routers.

C. CORE ROUTER CONFIGURATION

The core routers must be able to determine an appropriate PHB for each packet based only on the DSCP value that was assigned at the network edge. This requires the router to define appropriate classes to which packets are assigned according to their DSCP values. Next, the router must map the classes to PHB policies that carry out the desired forwarding, shaping, or policing behavior. Lastly, the router is configured to implement the policy on the outbound interface.

1. PHB Class Assignment

The classes that are created at the core routers must be defined based on DSCP valued. The core router does not know which classes were established at the edge router. It must, therefore, be able to create DiffServ classes independent of the edge classes. Based on the Cisco standard classes it is possible to define one EF class, four AF classes, and one DF class. For the example that is being examined, EF traffic will be contained in the platinum class. The example only needs three classes for AF traffic: gold, silver, and bronze. DF traffic will be contained in the best-effort class. The class maps are defined using all of the default DSCP values for that class, even those DSCPs that are not currently being used. This recommended practice allows for a consistent PHB class map to be maintained when the network requirements might change to add or delete specific marking policies. The same class map definitions should be used by all core routers. In the laboratory network these routers are defined to be ARFOR, ARMY_BGDE, MARFOR, and MAR_REGT. For a robust solution it would be desirable for the JTF router to also implement the same configuration. This is omitted in the laboratory network to limit the scope of the overall solution. The output nodes will not route traffic through JTF for testing purposes. All of the core routers implement the class maps shown in Figure 17.

```
class-map match-all platinum
  match ip dscp 46
class-map match-all gold
  match ip dscp 10 12 14
class-map match-all silver
  match ip dscp 18 20 22
class-map match-all bronze
  match ip dscp 26 28 30
class-map match-all best-effort
  match ip dscp 0
```

Figure 17. PHB Class Map for Core Routers.

2. PHB Policy Definition

The policy map definitions on the core routers associate PHBs with the classes that have been defined. For the platinum class a standard priority providing a minimum bandwidth value of 256 KBps is defined. The gold class is allocated 35 percent of the available bandwidth and enacts WRED based on the DSCP values of packets in the queue. The silver and bronze classes are each assigned 20 percent of the available bandwidth and implement DSCP-based WRED. The best-effort class polices traffic to an average rate of 56 KBps and a normal burst and maximum burst limit of 1750 Bytes. It transmits conforming packets and drops others. The same policy map is created on each core router as seen in Figure 18.

```
policy-map JMNO
  class platinum
    priority 256
  class gold
    bandwidth percent 35
    random-detect dscp-based
  class silver
    bandwidth percent 20
    random-detect dscp-based
  class bronze
    bandwidth percent 20
    random-detect dscp-based
  class best-effort
    police 56000 1750 1750 conform-action transmit
exceed-action drop violate-action drop
```

Figure 18. PHB Policy Map for Core Routers.

3. PHB Policy Assignment to Interface

Each interface of a core router that sends traffic outside of its local network should apply the PHB policies. Different approaches can be used based on network behavior preferences. For example, a military Service branch might

decide to only utilize the PHB policies when traffic is leaving for another Service's network. For the laboratory network the PHB policies are implemented on all outbound interfaces that exit the BGP AS of the router. The interfaces and the service policy statements are shown in Figure 19.

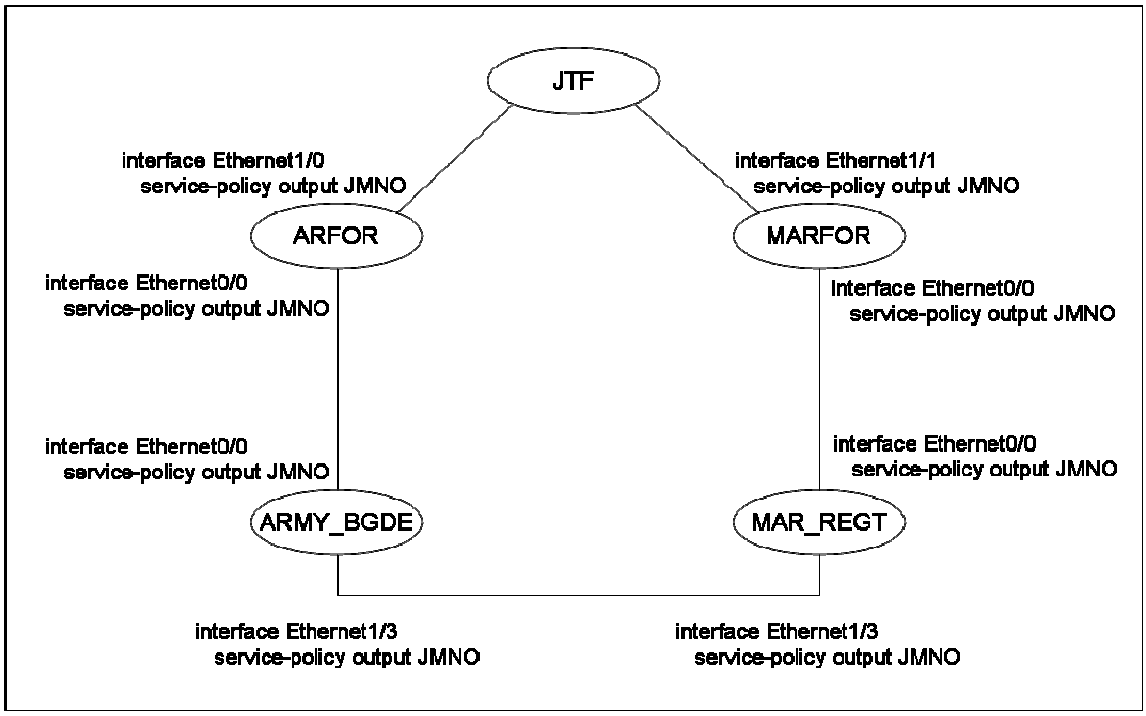

Figure 19. PHB Policy Assignments for Core Router Interfaces.

D. TEST RESULTS

With the DiffServ structure configured on the network routers, it is possible for the resulting behavior to be examined. To accomplish this, server processes are established on a host connected to the ARMY_BGDE router. These processes correspond with the port settings that are defined for the specific DiffServ classes. Additionally, client processes are created on hosts connected to routers on the Army portion of the network. This allows the marking and PHB policies to be examined prior to the traffic being routed to the server processes across the lateral link. For testing on the laboratory network, all of the traffic is

generated by a simulator that creates packet flows with a specified rate, size, and port assignment. The host connected to the MAR_BN1 router is assigned a local IP address, allowing its traffic to be marked as local unit traffic. The host connected to the MAR_BN2 router is assigned a non-local IP address, making its traffic be marked as originating from a mobile unit. Figure 20 depicts a simplified network diagram showing only those nodes that are critical to the testing scenario. It also shows the server and client processes established on each node.

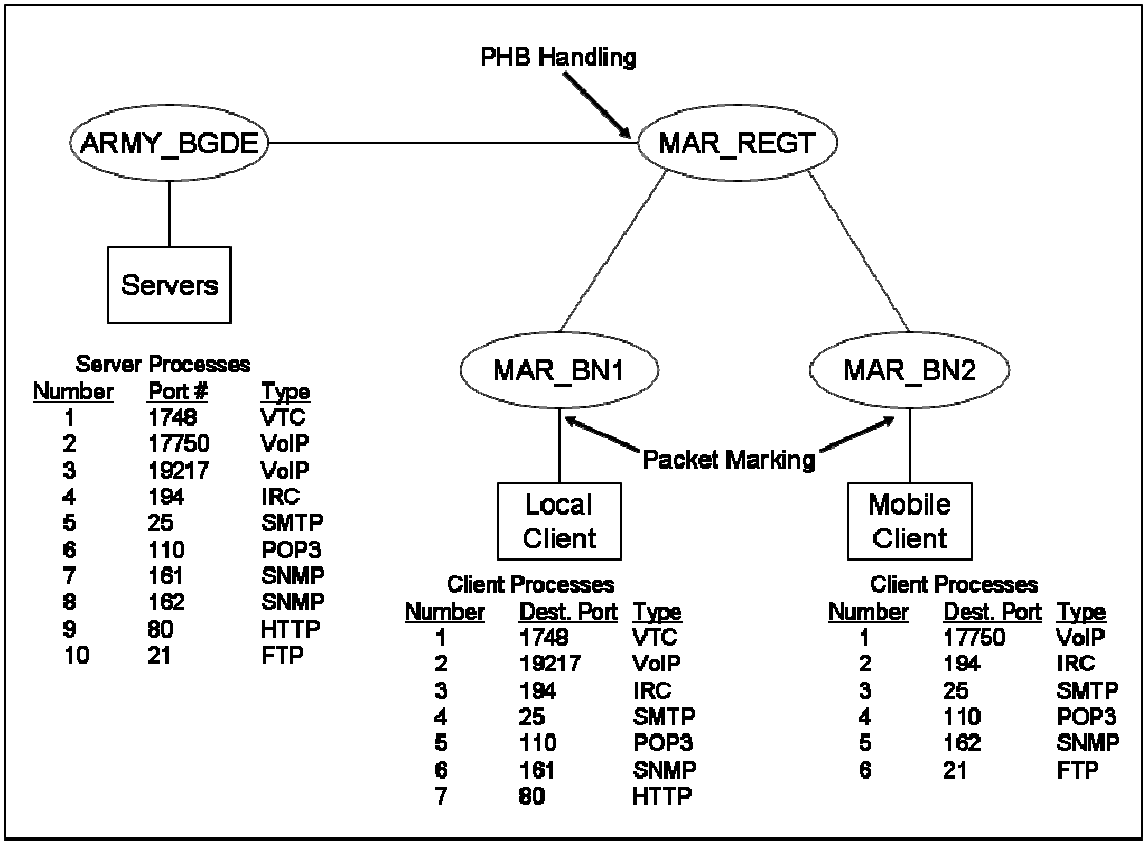

Figure 20. Testing Structure and Port Assignments.

Initial testing only implements the VTC traffic simulation from the local client to the server. The desired result is that the marking and PHB policies can be confirmed to be operating properly. This is accomplished by starting the server process and creating a client process to push traffic to the designated

port. Then it is possible to view the packet marking policy on the inbound interface. In this case, the desired interface is the FastEthernet1/0 interface on MAR_BN1. The policy-map is viewed by using the "show policy-map interface <*interface-number*>" command. The results from this command, as depicted in Figure 21, show that the VTC packets are properly being marked by the SETDSCP policy as belonging to the EF class.

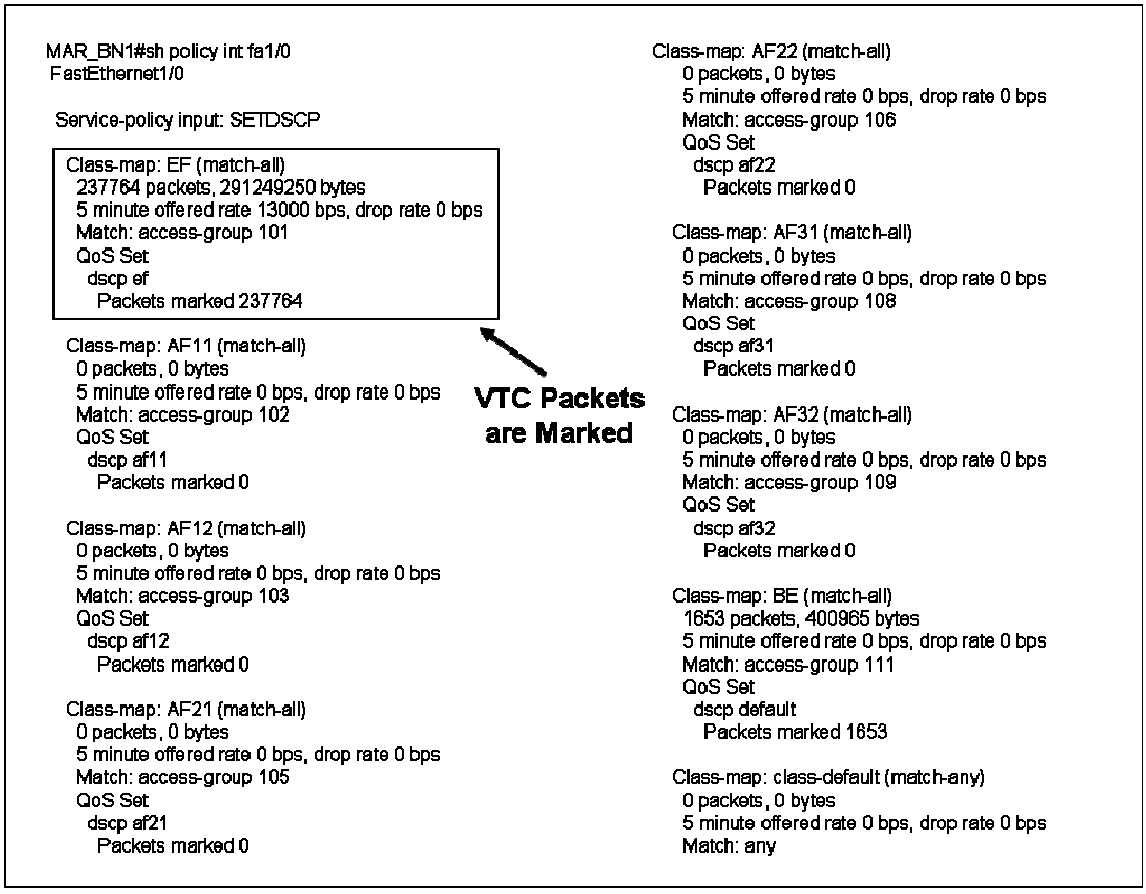

Figure 21. Test Results for VTC Packet Marking.

Next, the testing should confirm that the appropriate PHB policy is applied to the VTC packets as they are processed for transmission on the lateral link. For this portion of the test to be successful, the VTC traffic should be placed in the EF queue on the outbound interface (Ethernet1/3) on the MAR_REGT router. The same command format shown for viewing the marking policies is used to

examine the PHB policies. The policy-map that has been assigned to the particular interface is displayed. The output from this command when applied to an interface with a PHB policy-map is quite different from that seen for the marking policies. The show policy-map command displays queuing statistics and policy parameters for each PHB class. For brevity, only that portion of the show policy-map results relating to the VTC traffic flow are included in Figure 22. The complete result is included as Appendix K. The results show that VTC traffic is being placed into the platinum queue based on its DSCP marking for the EF class.

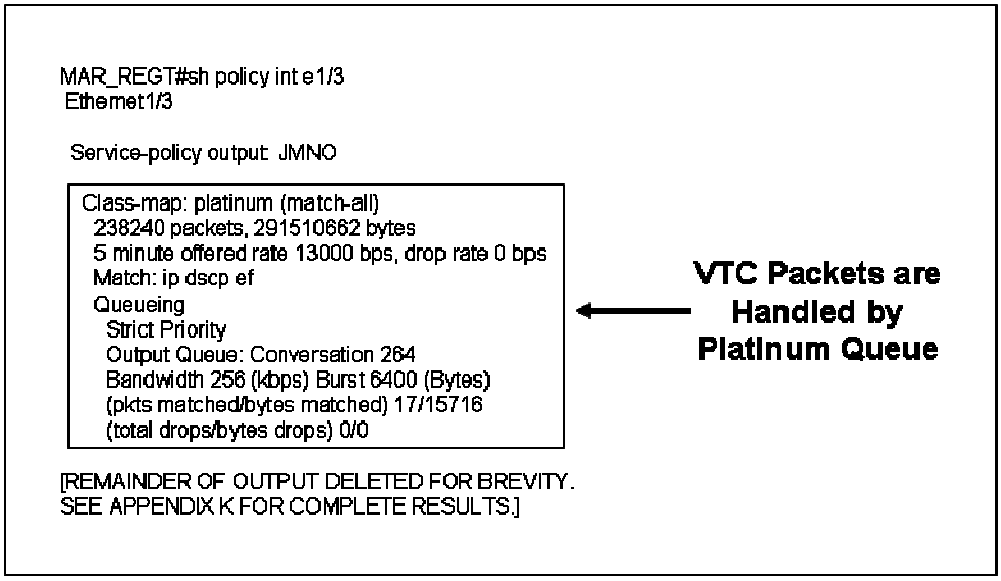

Figure 22. Test Result for VTC PHB Policy.

Confident that the VTC traffic is being properly marked and handled by the DiffServ network, it is possible to implement other simulated traffic flows that emulate the targeted processes. This will ensure that all of the classes are being marked properly and placed in the appropriate PHB queues. All of the processes for the server and two clients depicted in Figure 20 are created by the traffic generator. Each client process is assigned a specific packet size and transmit rate that it uses to send traffic to a specified port on the server. The settings for both the local and mobile clients are shown in Table 24.

Local Client

Client #	Port	Traffic Type	Interval (ms)	Packet Size (Bytes)	Output Rate (Bytes/sec)	Output Rate (bits/sec)	PHB Class	Class Marking
1	7648	VTC	1000	1500	1500	12000	Platinum	EF
2	19217	VoIP	50	220	4400	35200	Gold	AF11
3	194	IRC	100	250	2500	20000	Gold	AF12
4	25	SMTP	625	640	1024	8192	Bronze	AF31
5	110	POP3	1000	510	510	4080	Bronze	AF31
6	161	SNMP	600	300	500	4000	Bronze	AF31
7	80	HTTP	500	1500	3000	24000	Best-Effort	BE
				Total Output:	13434	107472		

Mobile Client

Client #	Port	Traffic Type	Interval (ms)	Packet Size (Bytes)	Output Rate (Bytes/sec)	Output Rate (bits/sec)	PHB Class	Class Marking
1	17750	VoIP	250	800	3200	25600	Silver	AF21
2	194	IRC	500	400	800	6400	Silver	AF22
3	25	SMTP	625	450	720	5760	Bronze	AF32
4	110	POP3	750	600	800	6400	Bronze	AF32
5	162	SNMP	1000	300	300	2400	Bronze	AF32
6	21	FTP	500	1000	2000	16000	Best-Effort	BE
				Total Output:	7820	62560		
				Total Lateral Link Traffic:	21254	170032		

Table 24. Traffic Generated for Test Network.

The flows are started on each client and begin sending traffic to the servers. The marking policies are then examined on each of the edge routers to ensure that the packets are being classified as intended. The local client's traffic being marked by the MAR_BN1 router is shown in Figure 23. The MAR_BN2 router is marking the traffic for the mobile client. Its policy-map is shown in Figure 24.

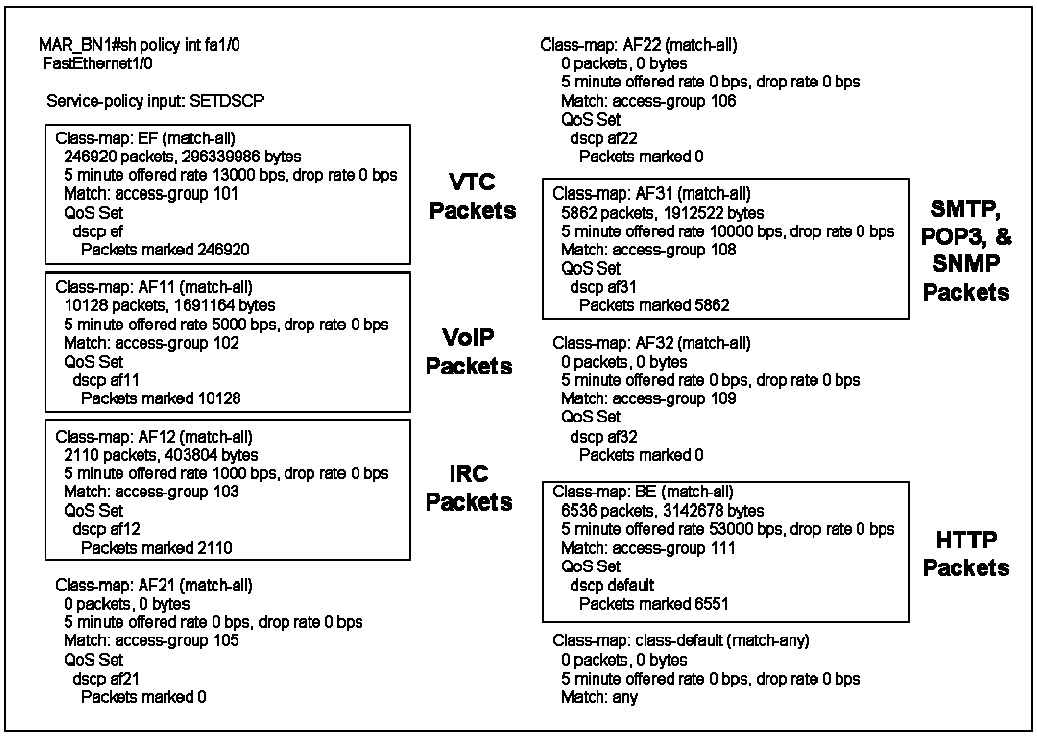

Figure 23. Packet Marking for Local Client Traffic.

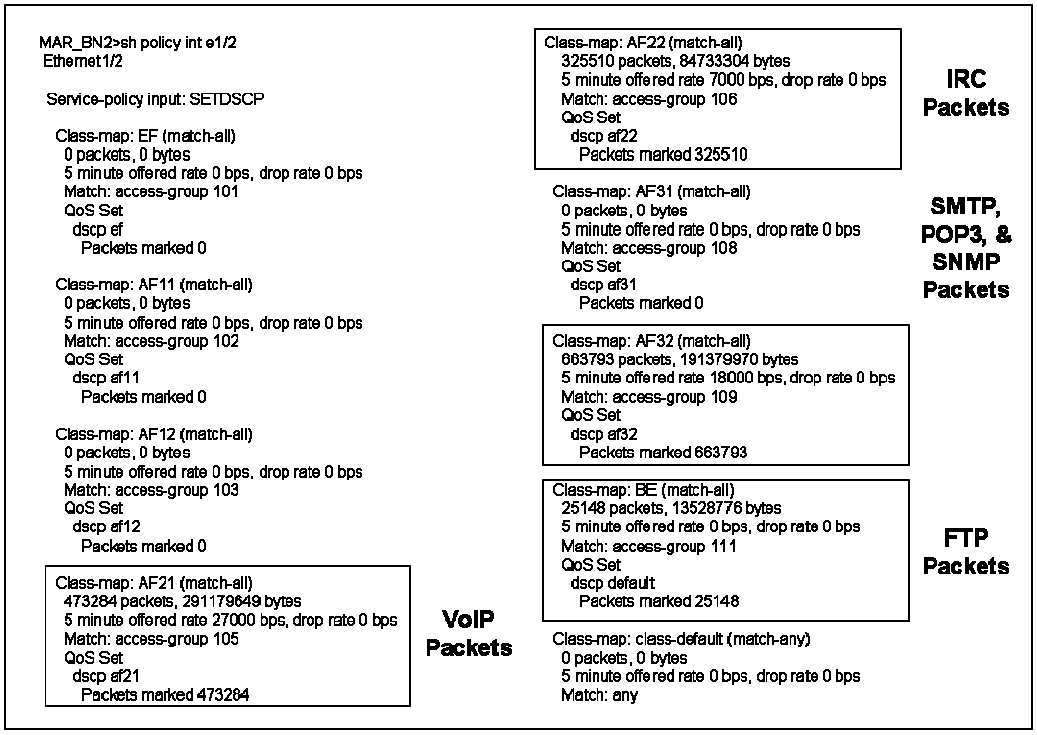

Figure 24. Packet Marking for Mobile Client Traffic.

The final check for this testing process is to determine whether or not the PHB policies are being implemented properly. As before, this is accomplished by looking at the policy-map for the outbound interface on the MAR_REGT core router. Due to the large size of the policy-map output, only select portions are displayed in Figure 25. The entire policy-map for interface Ethernet1/3 of MAR_REGT is included as Appendix L. The results clearly indicate that all of the traffic flows are being handled by the correct PHB queues. Thus, the DiffServ implementation is functioning as it was intended to do.

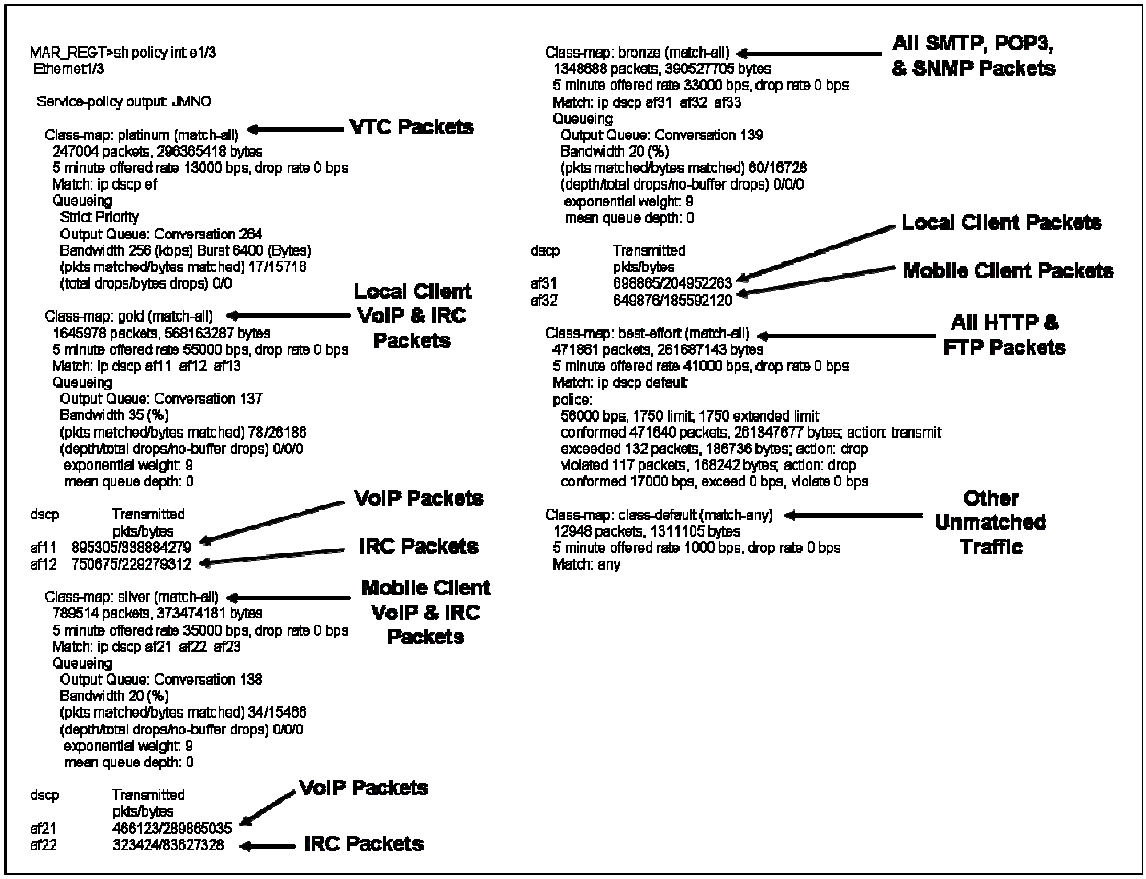

Figure 25. PHB Handling for all Test Traffic Classes.

The use of DiffServ allows QoS policies to be implemented in a scalable manner. The most important step in establishing DiffServ is determining the classes and PHBs that are desired. This is then implemented by establishing

edge routers and core routers that provide different functions within the DiffServ structure. Edge routers mark packets coming into the network through the use of access groups, class maps, and policy maps. These policies are then applied to the interface(s) where traffic enters the network. Core routers perform the DiffServ PHB policy administration. This is done by matching the DSCP value on each packet to a particular class, which is in turn mapped to a particular policy. The policy is applied to an outbound interface where the traffic leaves the local network. The overall QoS solution provides a means for critical time-sensitive traffic flows to be given priority handling, where desired. It also provides a means for particular units to ensure that the aggregate use of network resources by mobile units does not overwhelm the host unit's ability to perform its mission. Chapter V provides some specific conclusions and recommendations for future work regarding both network routing and QoS.

V. CONCLUSION AND RECOMMENDATIONS

This thesis seeks to provide JMNO with standardized network practices that can improve inter-Service connectivity. Within this overall goal, the focus is on providing specific routing solutions and an approach for QoS implementation. The criteria for evaluating these areas include ease of configuration, scalability, operational effectiveness, and simplicity. Not all of the solutions examined in this thesis successfully met all of the criteria. An evaluation of the routing and QoS solutions is provided below. This is followed by an examination of potential areas for future work on these topics.

A. ROUTING SOLUTIONS

The routing solutions examined in this thesis can be used to build templates for JMNO scenarios involving lateral Service connections and mobile unit routing. The solutions examined include the use of eBGP, iBGP, and DHCP for JMNO routers. The recommendations for JMNO use of these solutions is provided below.

1. eBGP Employment

It is recommended that eBGP be used to provide inter-AS connections (cf. Chapter III, Section A.1). It provides greater scalability and flexibility than the use of EIGRP to EIGRP redistribution. eBGP also does not rely on proprietary routing as part of its core solution. Business, education, government, and military networks already employ eBGP as the standard protocol for the Internet. JMNO should follow the same practice. It is recommended that eBGP be used for lateral link connections and for mobile units to connect to a different AS.

Although the BGP protocol is very complex, the proposed JMNO solution only implements those portions of the protocol that are actually needed. By taking a minimalist approach, it is possible to improve the simplicity and ease of

configuration factors for the protocol's use. One optional eBGP feature that is recommended for use is the ability to define and use filters based on the AS path. This can be used effectively to control the advertisement of routes learned from particular neighboring ASs. The employment of this technique is not particularly complicated and provides a solution to a specific JMNO concern.

2. iBGP Employment

In addition to the use of eBGP, it is recommended that iBGP be used for JMNO networks (cf. Chapter III, Section A.2). The use of iBGP within an AS allows nodes to exchange routing information with one another. Any interior routing protocol, including the use of static routes, can be used to exchange iBGP updates. The use of loopback addresses to establish iBGP neighbor relationships is strongly recommended. This allows the relationship to be maintained as long as a path can be found between the routers, even if a physical interface changes its address. For mobile units this is particularly important because iBGP allows them to maintain their membership in the home AS and have updates flow across the network.

3. DHCP Conclusions

Although DHCP is capable of providing automated IP address assignment, it is not recommended for use by JMNO for router port configurations (cf. Chapter III, Section B). The time and effort required to configure the DHCP server and client functions do not seem to be offset by an equivalent reduction of effort upon connecting a mobile host to a remote network. Even with the DHCP implementation used in this thesis, the mobile unit and connecting host routers must still manually configure eBGP (preferred method) or assign static routes. It does not require much additional effort to assign specific IP addresses to the appropriate interfaces. When the units coordinate their eBGP settings, they can also coordinate the IP assignments. Should a method for automating the routing updates be discovered, then the DHCP solution might

be beneficial enough to justify its use. This would allow the mobile unit to receive its IP address and routing configuration updates without any manual intervention.

B. QUALITY OF SERVICE SOLUTIONS

QoS allows networks to provide different levels of service based on specified characteristics of the traffic. JMNO can use QoS to ensure critical time-sensitive applications are able to perform as required. It also can provide differentiation between traffic generated by host units and mobile units. This allows JMNO to provide assurances that a host Service network will not be overwhelmed by traffic from mobile units. The recommended QoS approach is DiffServ. This provides greater flexibility and scalability than IntServ. It also provides improved bandwidth efficiency and service guarantees than best effort service. The use of DiffServ classes and the application of policies are discussed below.

1. Use of DiffServ Classes

It is recommended that DiffServ be implemented with the minimum number of classes required to achieve the operational requirements (cf. Chapter IV, Section A). Not all traffic requires the service guarantees provided by DiffServ. By minimizing the number of classes, the simplicity and effectiveness of DiffServ can be increased. Additionally, it is recommended that EF only be employed for critical applications that would otherwise be assigned statically reserved bandwidth. The example of a VTC application that normally uses a circuit switching approach is a valid case where EF might be used. DiffServ allows the application to receive a bandwidth guarantee when it is active, but allows other applications to use the bandwidth when the VTC is not running. Lastly, it is recommended that the default DiffServ classes and DSCP values be used on Cisco routers. This allows for greater simplicity in the structure and makes solutions more portable and adaptable to new requirements.

2. Application of DiffServ Policies

The DiffServ policies discussed in this thesis include traffic classification and PHB policies. How these policies are applied affects the overall effectiveness of the DiffServ implementation. It is recommended that traffic classification policies be applied as close to the network edge as possible (cf. Chapter IV, Section B). Generally, this would be any interface that connects to a LAN segment. It is not recommended, however, that mobile units be permitted to classify traffic. This is because policies for marking packets can vary between ASs. It is preferred, therefore, that the first interface of the DiffServ network that receives traffic from a mobile unit perform the traffic classification.

The application of PHB policies can vary greatly depending on the particular network being used. For most networks, it is recommended that the PHB policies be applied to interfaces where traffic exits the local AS (cf. Chapter IV, Section C). Generally, the interior links within an AS do not require DiffServ controls because they are high-bandwidth wired connections. It is generally on the lower-bandwidth links between ASs that congestion occurs. The actual network structure must be examined, though, to determine precisely where the PHB policies will be most effective.

C. FUTURE WORK

This thesis provides a starting point for JMNO routing and QoS solutions. Additional research remains open for investigation. Some of the potential areas for further study are presented below.

1. Automated Routing Updates

The proposed solution for mobile user connections requires manual configuration changes be made by both the host unit and the mobile unit. It may be possible to create a procedure that would allow this process to be automated. It would be desirable for the unit to be able to connect via a physical or wireless

interface without manual intervention. The mobile unit's interface could be assigned an IP address through DHCP. Then the routers might be able to exchange configuration information telling the other of its BGP AS number. This information could be used to automatically generate eBGP neighbor statements in the configuration files.

2. JMNO Application Requirements

It is currently not possible to provide a standard template for DiffServ structure because the actual applications employed by JMNO users and their bandwidth requirements are not known. It would be extremely useful for the final JMNO document to contain descriptions of the most common applications that might require DiffServ classification and handling. This should include the ports used by the application and its general bandwidth requirements. This information could be used to provide some standardized configuration examples that might be employed in DiffServ networks.

3. Operational Priority Classification

The DiffServ structure in this thesis provides mechanisms for classifying traffic based on the traffic type and source network. It might be desirable for military traffic to be classified by its operational priority as well. The tactical telephone system provides this type of service by allowing the user to provide a precedence value (Flash-Override, Flash, Immediate, etc.) before dialing. A similar structure might allow data traffic to receive priority handling when it is operationally needed. Any such structure would likely require a means for supervising its use to ensure it is not abused. If every packet is sent with the highest precedence value, then no packet will receive priority handling. Because the scope of this problem is so great, it might be necessary to limit its use to specific applications.

4. BGP Distribution of DiffServ Policies

A final area that might merit further investigation is the distribution of DiffServ policies through BGP. Within an AS it is generally desired that all routers implement the same DiffServ policies. The manual configuration process, however, provides a large amount of room for user error. BGP can provide a centralized distribution of these policies to ensure they are standardized throughout the AS [7]. It might be possible to create the policies on one router and have them distributed appropriately. Traffic classification policies could be distributed to all edge routers and the PHB policies could be sent to the core routers. Changes to the policies could be made on the central router and shared with its peers.

VI. APPENDICES

A. DISN ROUTER CONFIGURATION

```
Current configuration : 980 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname DISN
!
boot-start-marker
boot-end-marker
!
enable password jmno
!
no aaa new-model
ip subnet-zero
ip cef
!
interface Loopback0
 ip address 192.168.16.1 255.255.255.255
!
interface Ethernet0/0
 bandwidth 4096
 ip address 192.168.17.1 255.255.255.0
 half-duplex
!
interface FastEthernet1/0
 ip address 192.168.16.9 255.255.255.248
 duplex auto
 speed auto
!
router eigrp 16
 redistribute bgp 60000
 network 192.168.16.0 0.0.0.7
 network 192.168.16.8 0.0.0.7
 no auto-summary
!
router bgp 60000
 no synchronization
```

```
 bgp log-neighbor-changes
 network 192.168.17.0
 redistribute eigrp 16
 neighbor 192.168.17.2 remote-as 61000
 neighbor 192.168.17.2 description JTF
 neighbor 192.168.17.2 soft-reconfiguration inbound
 no auto-summary
!
no ip http server
ip classless
!
line con 0
line aux 0
line vty 0 4
 password jmno
 login
!
end
```

B. JTF ROUTER CONFIGURATION

```
Current configuration : 1420 bytes
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname JTF
!
enable password jmno
!
memory-size iomem 15
ip subnet-zero
!
interface Loopback0
 ip address 192.168.18.1 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.19.1 255.255.255.0
!
interface Ethernet1/0
 ip address 192.168.17.2 255.255.255.0
!
interface Ethernet1/1
 ip address 192.168.21.1 255.255.255.0
```

```
!
interface Ethernet1/2
 no ip address
!
interface Ethernet1/3
 ip address 192.168.18.9 255.255.255.248
!
router eigrp 18
 redistribute bgp 61000
 network 192.168.18.0 0.0.0.7
 network 192.168.18.8 0.0.0.7
 no auto-summary
 no eigrp log-neighbor-changes
!
router bgp 61000
 no synchronization
 bgp log-neighbor-changes
 network 192.168.17.0
 network 192.168.18.0
 network 192.168.19.0
 network 192.168.21.0
 redistribute eigrp 18
 neighbor 192.168.17.1 remote-as 60000
 neighbor 192.168.17.1 description DISN
 neighbor 192.168.17.1 soft-reconfiguration inbound
 neighbor 192.168.19.2 remote-as 65010
 neighbor 192.168.19.2 description ARFOR
 neighbor 192.168.19.2 soft-reconfiguration inbound
 neighbor 192.168.21.2 remote-as 65020
 neighbor 192.168.21.2 description MARFOR
 neighbor 192.168.21.2 soft-reconfiguration inbound
 no auto-summary
!
ip classless
no ip http server
!
line con 0
 transport input none
line aux 0
line vty 0 4
 password jmno
 login
!
end
```

C. ARFOR ROUTER CONFIGURATION

```
Current configuration : 2210 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname ARFOR
!
enable password jmno
!
memory-size iomem 10
ip subnet-zero
!
!
ip dhcp excluded-address 192.168.39.1
ip dhcp excluded-address 192.168.39.5
!
ip dhcp pool ARFOR_E1/1
   network 192.168.39.0 255.255.255.252
   default-router 192.168.39.1
!
ip dhcp pool ARFOR_E1/2
   network 192.168.39.4 255.255.255.252
   default-router 192.168.39.5
!
!
!
class-map match-all gold
  match ip dscp af11  af12  af13
class-map match-all bronze
  match ip dscp af31  af32  af33
class-map match-all platinum
  match ip dscp ef
class-map match-all silver
  match ip dscp af21  af22  af23
class-map match-all best-effort
  match ip dscp default
!
!
policy-map JMNO
  class platinum
    priority 256
```

```
  class gold
   bandwidth percent 35
   random-detect dscp-based
  class silver
   bandwidth percent 20
   random-detect dscp-based
  class bronze
   bandwidth percent 20
   random-detect dscp-based
  class best-effort
     police 65000 1750 1750 conform-action transmit exceed-action drop
violate-a
ction drop
!
!
!
interface Loopback0
 ip address 192.168.32.1 255.255.255.255
!
interface Ethernet0/0
 bandwidth 512
 ip address 192.168.33.1 255.255.255.0
 service-policy output JMNO
 half-duplex
!
interface Ethernet1/0
 bandwidth 3608
 ip address 192.168.19.2 255.255.255.0
 service-policy output JMNO
 half-duplex
!
interface Ethernet1/1
 ip address 192.168.39.1 255.255.255.252
 half-duplex
!
interface Ethernet1/2
 ip address 192.168.39.5 255.255.255.252
 half-duplex
!
interface Ethernet1/3
 ip address 192.168.32.9 255.255.255.248
 half-duplex
!
router eigrp 32
 redistribute bgp 65010
```

```
 network 192.168.32.0 0.0.0.7
 network 192.168.32.8 0.0.0.7
 no auto-summary
!
router bgp 65010
 no synchronization
 bgp log-neighbor-changes
 network 192.168.19.0
 network 192.168.33.0
 redistribute eigrp 32
 neighbor 192.168.19.1 remote-as 61000
 neighbor 192.168.19.1 soft-reconfiguration inbound
 neighbor 192.168.33.2 remote-as 65011
 neighbor 192.168.33.2 soft-reconfiguration inbound
 no auto-summary
!
ip classless
no ip http server
!
!
line con 0
line aux 0
line vty 0 4
 password jmno
 login
!
end
```

D. ARMY_BGDE ROUTER CONFIGURATION

```
Current configuration : 2763 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname ARMY_BGDE
!
enable password jmno
!
ip subnet-zero
!
!
ip dhcp excluded-address 192.168.39.13
ip dhcp excluded-address 192.168.39.17
```

```
!
ip dhcp pool ARMY_BGDE_E1/0
   network 192.168.39.12 255.255.255.252
   default-router 192.168.39.13
!
ip dhcp pool ARMY_BGDE_E1/1
   network 192.168.39.16 255.255.255.252
   default-router 192.168.39.17
!
call rsvp-sync
!
!
!
!
!
!
!
class-map match-all gold
  match ip dscp af11  af12  af13
class-map match-all bronze
  match ip dscp af31  af32  af33
class-map match-all platinum
  match ip dscp ef
class-map match-all silver
  match ip dscp af21  af22  af23
class-map match-all best-effort
  match ip dscp default
!
!
policy-map JMNO
  class platinum
    priority 256
  class gold
   bandwidth percent 35
   random-detect dscp-based
  class silver
   bandwidth percent 20
   random-detect dscp-based
  class bronze
   bandwidth percent 20
   random-detect dscp-based
  class best-effort
     police 56000 1750 1750 conform-action transmit exceed-action drop
violate-a
ction drop
```

```
!
!
!
interface Loopback0
 ip address 192.168.34.1 255.255.255.255
!
interface Ethernet0/0
 bandwidth 512
 ip address 192.168.33.2 255.255.255.0
 service-policy output JMNO
 half-duplex
!
interface Ethernet1/0
 bandwidth 256
 ip address 192.168.39.13 255.255.255.252
 half-duplex
!
interface Ethernet1/1
 bandwidth 256
 ip address 192.168.39.17 255.255.255.252
 half-duplex
!
interface Ethernet1/2
 ip address 192.168.34.9 255.255.255.248
 half-duplex
!
interface Ethernet1/3
 bandwidth 512
 ip address 192.168.23.1 255.255.255.0
 service-policy output JMNO
 half-duplex
!
router eigrp 321
 redistribute bgp 65011
 network 192.168.34.0 0.0.0.7
 network 192.168.34.8 0.0.0.7
 network 192.168.39.12 0.0.0.3
 network 192.168.39.16 0.0.0.3
 network 192.168.32.0 0.0.31.255
 no auto-summary
!
router bgp 65011
 no synchronization
 bgp log-neighbor-changes
 network 192.168.23.0
```

```
 network 192.168.33.0
 network 192.168.39.12 mask 255.255.255.252
 network 192.168.39.16 mask 255.255.255.252
 redistribute eigrp 321
 neighbor 192.168.23.2 remote-as 65021
 neighbor 192.168.23.2 soft-reconfiguration inbound
 neighbor 192.168.33.1 remote-as 65010
 neighbor 192.168.33.1 soft-reconfiguration inbound
 neighbor 192.168.33.1 filter-list 1 out
 neighbor 192.168.36.1 remote-as 65011
 neighbor 192.168.36.1 update-source Loopback0
 neighbor 192.168.38.1 remote-as 65011
 neighbor 192.168.38.1 update-source Loopback0
 no auto-summary
!
ip classless
no ip http server
ip as-path access-list 1 deny _65021_
ip as-path access-list 1 permit .*
!
!
dial-peer cor custom
!
!
!
!
line con 0
line aux 0
line vty 0 4
 password jmno
 login
!
end
```

E. ARMY_BN1 ROUTER CONFIGURATION

```
Current configuration : 2665 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname ARMY_BN1
!
boot-start-marker
```

```
boot-end-marker
!
enable password jmno
!
no network-clock-participate slot 1
no network-clock-participate wic 0
no aaa new-model
ip subnet-zero
no ip cef
!
!
!
ip audit po max-events 100
!
!
!
!
!
!
!
!
!
!
!
!
!
!
!
class-map match-all EF
  match access-group 101
class-map match-all AF12
  match access-group 103
class-map match-all AF21
  match access-group 105
class-map match-all AF31
  match access-group 108
class-map match-all AF32
  match access-group 109
class-map match-all AF11
  match access-group 102
class-map match-all AF22
  match access-group 106
class-map match-all BE
  match access-group 111
!
!
```

```
policy-map SETDSCP
  class EF
   set ip dscp ef
  class AF11
   set ip dscp af11
  class AF12
   set ip dscp af12
  class AF21
   set ip dscp af21
  class AF22
   set ip dscp af22
  class AF31
   set ip dscp af31
  class AF32
   set ip dscp af32
  class BE
   set ip dscp default
!
!
!
!
!
!
interface Loopback0
 ip address 192.168.36.1 255.255.255.255
!
interface FastEthernet0/0
 ip address dhcp client-id FastEthernet0/0
 duplex auto
 speed 10
!
interface FastEthernet0/1
 ip address 192.168.36.9 255.255.255.248
 duplex auto
 speed 10
 service-policy input SETDSCP
!
router eigrp 321
 redistribute bgp 65011
 network 192.168.36.0 0.0.0.7
 network 192.168.36.8 0.0.0.7
 network 192.168.39.12 0.0.0.3
 network 192.168.32.0 0.0.31.255
 no auto-summary
!
```

```
router bgp 65011
 no synchronization
 bgp log-neighbor-changes
 network 192.168.36.0
 neighbor 192.168.34.1 remote-as 65011
 neighbor 192.168.34.1 update-source Loopback0
 neighbor 192.168.38.1 remote-as 65011
 neighbor 192.168.38.1 update-source Loopback0
 no auto-summary
!
no ip http server
no ip http secure-server
ip classless
!
!
access-list 101 permit udp any any eq 24032
access-list 101 permit udp any any range 7648 7649
access-list 102 permit udp 192.168.32.0 0.0.31.255 any range 16384
32768
access-list 103 permit udp 192.168.32.0 0.0.31.255 any eq 194
access-list 105 permit udp any any range 16384 32768
access-list 106 permit udp any any eq 194
access-list 108 permit tcp 192.168.32.0 0.0.31.255 any eq smtp
access-list 108 permit tcp 192.168.32.0 0.0.31.255 any eq pop3
access-list 108 permit tcp 192.168.32.0 0.0.31.255 any range 161 162
access-list 109 permit tcp any any eq smtp
access-list 109 permit tcp any any eq pop3
access-list 109 permit tcp any any range 161 162
access-list 111 permit ip any any
no cdp log mismatch duplex
!
!
!
!
!
!
line con 0
line aux 0
line vty 0 4
 password jmno
 login
!
!
end
```

F. ARMY_BN2 ROUTER CONFIGURATION

```
Current configuration : 2417 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname "ARMY_BN2"
!
enable password jmno
!
memory-size iomem 15
ip subnet-zero
ip cef
!
!
!
!
class-map match-all EF
  match access-group 101
class-map match-all AF12
  match access-group 103
class-map match-all AF21
  match access-group 105
class-map match-all AF31
  match access-group 108
class-map match-all AF32
  match access-group 109
class-map match-all AF11
  match access-group 102
class-map match-all AF22
  match access-group 106
class-map match-all BE
  match access-group 111
!
!
policy-map SETDSCP
  class EF
   set ip dscp ef
  class AF11
   set ip dscp af11
  class AF12
   set ip dscp af12
```

```
  class AF21
   set ip dscp af21
  class AF22
   set ip dscp af22
  class AF31
   set ip dscp af31
  class AF32
   set ip dscp af32
  class BE
   set ip dscp default
!
!
!
!
interface Loopback0
 ip address 192.168.38.1 255.255.255.255
!
interface Ethernet0/0
 bandwidth 256
 ip address dhcp client-id Ethernet0/0
 half-duplex
!
interface FastEthernet1/0
 ip address 192.168.38.9 255.255.255.248
 speed 10
 half-duplex
 service-policy input SETDSCP
!
router eigrp 321
 redistribute bgp 65011
 network 192.168.38.0 0.0.0.7
 network 192.168.38.8 0.0.0.7
 network 192.168.39.8 0.0.0.7
 network 192.168.32.0 0.0.31.255
 no auto-summary
!
router bgp 65011
 no synchronization
 bgp log-neighbor-changes
 network 192.168.38.0
 neighbor 192.168.34.1 remote-as 65011
 neighbor 192.168.34.1 update-source Loopback0
 neighbor 192.168.36.1 remote-as 65011
 neighbor 192.168.36.1 update-source Loopback0
 no auto-summary
```

```
!
ip classless
no ip http server
!
access-list 101 permit udp any any range 7648 7649
access-list 101 permit udp any any eq 24032
access-list 102 permit udp 192.168.32.0 0.0.31.255 any range 16384
32768
access-list 103 permit udp 192.168.32.0 0.0.31.255 any eq 194
access-list 105 permit udp any any range 16384 32768
access-list 106 permit udp any any eq 194
access-list 108 permit tcp 192.168.32.0 0.0.31.255 any eq smtp
access-list 108 permit tcp 192.168.32.0 0.0.31.255 any eq pop3
access-list 108 permit tcp 192.168.32.0 0.0.31.255 any range 161 162
access-list 109 permit tcp any any eq smtp
access-list 109 permit tcp any any eq pop3
access-list 109 permit tcp any any range 161 162
access-list 111 permit ip any any
!
line con 0
line aux 0
line vty 0 4
 password jmno
 login
!
end
```

G. MARFOR ROUTER CONFIGURATION

```
Current configuration : 2362 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname MARFOR
!
boot-start-marker
boot-end-marker
!
enable password jmno
!
no aaa new-model
ip subnet-zero
ip cef
```

```
!
!
ip dhcp excluded-address 192.168.71.1
ip dhcp excluded-address 192.168.71.5
!
ip dhcp pool MARFOR_E1/0
   network 192.168.71.0 255.255.255.252
   default-router 192.168.71.1
!
ip dhcp pool MARFOR_E1/2
   network 192.168.71.4 255.255.255.252
   default-router 192.168.71.5
!
!
!
!
class-map match-all gold
  match ip dscp af11  af12  af13
class-map match-all bronze
  match ip dscp af31  af32  af33
class-map match-all platinum
  match ip dscp ef
class-map match-all silver
  match ip dscp af21  af22  af23
class-map match-all best-effort
  match ip dscp default
!
!
policy-map JMNO
  class platinum
   priority 256
  class gold
   bandwidth percent 35
   random-detect dscp-based
  class silver
   bandwidth percent 20
   random-detect dscp-based
  class bronze
   bandwidth percent 15
   random-detect dscp-based
  class best-effort
   police 56000 1750 1750 conform-action set-dscp-transmit 0 conform-
action tran
smit  exceed-action drop  violate-action drop
!
```

```
!
!
interface Loopback0
 ip address 192.168.64.1 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.21.2 255.255.255.0
 half-duplex
 service-policy output JMNO
!
interface Ethernet1/0
 ip address 192.168.71.1 255.255.255.252
 half-duplex
!
interface Ethernet1/1
 ip address 192.168.65.1 255.255.255.0
 half-duplex
 service-policy output JMNO
!
interface Ethernet1/2
 ip address 192.168.71.5 255.255.255.252
 half-duplex
!
interface Ethernet1/3
 ip address 192.168.64.9 255.255.255.248
 half-duplex
!
router eigrp 64
 redistribute bgp 65020
 network 192.168.64.1 0.0.0.0
 network 192.168.64.8 0.0.0.7
 no auto-summary
!
router bgp 65020
 no synchronization
 bgp log-neighbor-changes
 network 192.168.21.0
 network 192.168.65.0
 network 192.168.71.0 mask 255.255.255.252
 network 192.168.71.4 mask 255.255.255.252
 redistribute eigrp 64
 neighbor 192.168.21.1 remote-as 61000
 neighbor 192.168.21.1 soft-reconfiguration inbound
 neighbor 192.168.65.2 remote-as 65021
 neighbor 192.168.65.2 soft-reconfiguration inbound
```

```
 no auto-summary
!
no ip http server
ip classless
!
!
!
line con 0
line aux 0
line vty 0 4
 password jmno
 login
!
!
end
```

H. MAR_REGT ROUTER CONFIGURATION

```
Current configuration : 2781 bytes
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname MAR_REGT
!
enable password jmno
!
ip subnet-zero
!
!
ip dhcp excluded-address 192.168.71.13
ip dhcp excluded-address 192.168.71.17
!
ip dhcp pool MARFOR_E1/0
   network 192.168.71.12 255.255.255.252
   default-router 192.168.71.13
!
ip dhcp pool MARFOR_E1/1
   network 192.168.71.16 255.255.255.252
   default-router 192.168.71.17
!
no call rsvp-sync
!
!
```

```
!
!
!
!
!
class-map match-all gold
  match ip dscp af11  af12  af13
class-map match-all bronze
  match ip dscp af31  af32  af33
class-map match-all platinum
  match ip dscp ef
class-map match-all silver
  match ip dscp af21  af22  af23
class-map match-all best-effort
  match ip dscp default
!
!
policy-map JMNO
  class platinum
    priority 256
  class gold
   bandwidth percent 35
   random-detect dscp-based
  class silver
   bandwidth percent 20
   random-detect dscp-based
  class bronze
   bandwidth percent 20
   random-detect dscp-based
  class best-effort
     police 56000 1750 1750 conform-action transmit exceed-action drop
violate-a
ction drop
!
!
!
interface Loopback0
 ip address 192.168.66.1 255.255.255.255
!
interface Ethernet0/0
 ip address 192.168.65.2 255.255.255.0
 service-policy output JMNO
 no ip mroute-cache
 half-duplex
!
```

```
interface Ethernet1/0
 ip address 192.168.71.13 255.255.255.252
 no ip mroute-cache
 half-duplex
!
interface Ethernet1/1
 ip address 192.168.71.17 255.255.255.252
 no ip mroute-cache
 half-duplex
!
interface Ethernet1/2
 ip address 192.168.66.9 255.255.255.248
 no ip mroute-cache
 half-duplex
!
interface Ethernet1/3
 ip address 192.168.23.2 255.255.255.0
 service-policy output JMNO
 no ip mroute-cache
 half-duplex
!
router eigrp 641
 redistribute bgp 65021
 network 192.168.66.0 0.0.0.7
 network 192.168.66.8 0.0.0.7
 network 192.168.67.0
 network 192.168.69.0
 network 192.168.64.0 0.0.31.255
 no auto-summary
!
router bgp 65021
 no synchronization
 bgp log-neighbor-changes
 network 192.168.23.0
 network 192.168.65.0
 network 192.168.71.12 mask 255.255.255.252
 network 192.168.71.16 mask 255.255.255.252
 redistribute eigrp 641
 neighbor 192.168.23.1 remote-as 65011
 neighbor 192.168.23.1 soft-reconfiguration inbound
 neighbor 192.168.65.1 remote-as 65020
 neighbor 192.168.65.1 soft-reconfiguration inbound
 neighbor 192.168.65.1 filter-list 1 out
 neighbor 192.168.68.1 remote-as 65021
 neighbor 192.168.68.1 update-source Loopback0
```

```
 neighbor 192.168.70.1 remote-as 65021
 neighbor 192.168.70.1 update-source Loopback0
 no auto-summary
!
ip classless
no ip http server
ip as-path access-list 1 deny _65011_
ip as-path access-list 1 permit .*
!
!
dial-peer cor custom
!
!
!
!
line con 0
line aux 0
line vty 0 4
 password jmno
 login
!
end
```

I. MAR_BN1 ROUTER CONFIGURATION

```
Current configuration : 2442 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname MAR_BN1
!
boot-start-marker
boot-end-marker
!
enable password jmno
!
no aaa new-model
no ip subnet-zero
no ip cef
!
!
!
!
```

```
!
!
class-map match-all EF
  match access-group 101
class-map match-all AF12
  match access-group 103
class-map match-all AF21
  match access-group 105
class-map match-all AF31
  match access-group 108
class-map match-all AF32
  match access-group 109
class-map match-all AF11
  match access-group 102
class-map match-all AF22
  match access-group 106
class-map match-all BE
  match access-group 111
!
!
policy-map SETDSCP
  class EF
   set ip dscp ef
  class AF11
   set ip dscp af11
  class AF12
   set ip dscp af12
  class AF21
   set ip dscp af21
  class AF22
   set ip dscp af22
  class AF31
   set ip dscp af31
  class AF32
   set ip dscp af32
  class BE
   set ip dscp default
!
!
!
interface Loopback0
 ip address 192.168.68.1 255.255.255.255
!
interface Ethernet0/0
 bandwidth 256
```

```
 ip address dhcp client-id Ethernet0/0
 half-duplex
!
interface FastEthernet1/0
 ip address 192.168.68.9 255.255.255.248
 duplex auto
 speed 10
 service-policy input SETDSCP
!
router eigrp 641
 redistribute bgp 65021
 network 192.168.68.0 0.0.0.7
 network 192.168.68.8 0.0.0.7
 network 192.168.64.0 0.0.31.255
 no auto-summary
!
router bgp 65021
 no synchronization
 bgp log-neighbor-changes
 network 192.168.68.0
 neighbor 192.168.66.1 remote-as 65021
 neighbor 192.168.66.1 update-source Loopback0
 neighbor 192.168.70.1 remote-as 65021
 neighbor 192.168.70.1 update-source Loopback0
 no auto-summary
!
no ip http server
ip classless
!
!
access-list 101 permit udp any any range 7648 7649
access-list 101 permit udp any any eq 24032
access-list 102 permit udp 192.168.64.0 0.0.31.255 any range 16384
32768
access-list 103 permit udp 192.168.64.0 0.0.31.255 any eq 194
access-list 105 permit udp any any range 16384 32768
access-list 106 permit udp any any eq 194
access-list 108 permit tcp 192.168.64.0 0.0.31.255 any eq smtp
access-list 108 permit tcp 192.168.64.0 0.0.31.255 any eq pop3
access-list 108 permit tcp 192.168.64.0 0.0.31.255 any range 161 162
access-list 109 permit tcp any any eq smtp
access-list 109 permit tcp any any eq pop3
access-list 109 permit tcp any any range 161 162
access-list 111 permit ip any any
!
```

```
line con 0
line aux 0
line vty 0 4
 password jmno
 login
!
!
end
```

J. MAR_BN2 ROUTER CONFIGURATION

```
Current configuration : 2758 bytes
!
version 12.3
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname MAR_BN2
!
boot-start-marker
boot-end-marker
!
enable password jmno
!
no network-clock-participate slot 1
no network-clock-participate wic 0
no aaa new-model
ip subnet-zero
ip cef
!
!
!
!
!
!
class-map match-all EF
  match access-group 101
class-map match-all AF12
  match access-group 103
class-map match-all AF21
  match access-group 105
class-map match-all AF31
  match access-group 108
class-map match-all AF32
  match access-group 109
```

```
class-map match-all AF11
  match access-group 102
class-map match-all AF22
  match access-group 106
class-map match-all BE
  match access-group 111
!
!
policy-map SETDSCP
  class EF
   set ip dscp ef
  class AF11
   set ip dscp af11
  class AF12
   set ip dscp af12
  class AF21
   set ip dscp af21
  class AF22
   set ip dscp af22
  class AF31
   set ip dscp af31
  class AF32
   set ip dscp af32
  class BE
   set ip dscp default
!
!
!
interface Loopback0
 ip address 192.168.70.1 255.255.255.255
!
interface FastEthernet0/0
 no ip address
 shutdown
 speed 10
 half-duplex
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Ethernet1/0
 ip address 192.168.70.9 255.255.255.248
```

```
 half-duplex
 service-policy input SETDSCP
!
interface Ethernet1/1
 ip address dhcp client-id Ethernet1/1
 half-duplex
!
interface Ethernet1/2
 no ip address
 shutdown
 full-duplex
!
interface Ethernet1/3
 no ip address
 shutdown
 half-duplex
!
router eigrp 641
 redistribute bgp 65021
 network 192.168.70.0 0.0.0.7
 network 192.168.70.8 0.0.0.7
 network 192.168.64.0 0.0.31.255
 no auto-summary
!
router bgp 65021
 no synchronization
 bgp log-neighbor-changes
 network 192.168.70.0
 neighbor 192.168.66.1 remote-as 65021
 neighbor 192.168.66.1 update-source Loopback0
 neighbor 192.168.68.1 remote-as 65021
 neighbor 192.168.68.1 update-source Loopback0
 no auto-summary
!
no ip http server
ip classless
!
!
access-list 101 permit udp any any range 7648 7649
access-list 101 permit udp any any eq 24032
access-list 102 permit udp 192.168.64.0 0.0.31.255 any range 16384
32768
access-list 103 permit udp 192.168.64.0 0.0.31.255 any eq 194
access-list 105 permit udp any any range 16384 32768
access-list 106 permit udp any any eq 194
```

```
access-list 108 permit tcp 192.168.64.0 0.0.31.255 any eq smtp
access-list 108 permit tcp 192.168.64.0 0.0.31.255 any eq pop3
access-list 108 permit tcp 192.168.64.0 0.0.31.255 any range 161 162
access-list 109 permit tcp any any eq smtp
access-list 109 permit tcp any any eq pop3
access-list 109 permit tcp any any range 161 162
access-list 111 permit ip any any
!
line con 0
line aux 0
line vty 0 4
 password jmno
 login
!
!
end
```

K. PHB POLICY-MAP FOR VTC-ONLY TRAFFIC ON MAR_REGT

```
MAR_REGT#sh policy int e1/3
  Ethernet1/3

  Service-policy output: JMNO

    Class-map: platinum (match-all)
      238240 packets, 291510662 bytes
      5 minute offered rate 13000 bps, drop rate 0 bps
      Match: ip dscp ef
      Queueing
        Strict Priority
        Output Queue: Conversation 264
        Bandwidth 256 (kbps) Burst 6400 (Bytes)
        (pkts matched/bytes matched) 17/15716
        (total drops/bytes drops) 0/0

    Class-map: gold (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: ip dscp af11  af12  af13
      Queueing
        Output Queue: Conversation 265
        Bandwidth 35 (%)
        (pkts matched/bytes matched) 0/0
        (depth/total drops/no-buffer drops) 0/0/0
         exponential weight: 9
```

mean queue depth: 0

dscp	Transmitted pkts/bytes	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark prob
af11	0/0	0/0	0/0	32	40	1/10
af12	0/0	0/0	0/0	28	40	1/10
af13	0/0	0/0	0/0	24	40	1/10
af21	0/0	0/0	0/0	32	40	1/10
af22	0/0	0/0	0/0	28	40	1/10
af23	0/0	0/0	0/0	24	40	1/10
af31	0/0	0/0	0/0	32	40	1/10
af32	0/0	0/0	0/0	28	40	1/10
af33	0/0	0/0	0/0	24	40	1/10
af41	0/0	0/0	0/0	32	40	1/10
af42	0/0	0/0	0/0	28	40	1/10
af43	0/0	0/0	0/0	24	40	1/10
cs1	0/0	0/0	0/0	22	40	1/10
cs2	0/0	0/0	0/0	24	40	1/10
cs3	0/0	0/0	0/0	26	40	1/10
cs4	0/0	0/0	0/0	28	40	1/10
cs5	0/0	0/0	0/0	30	40	1/10
cs6	0/0	0/0	0/0	32	40	1/10
cs7	0/0	0/0	0/0	34	40	1/10
ef	0/0	0/0	0/0	36	40	1/10
rsvp	0/0	0/0	0/0	36	40	1/10
default	0/0	0/0	0/0	20	40	1/10

Class-map: silver (match-all)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: ip dscp af21 af22 af23
Queueing
Output Queue: Conversation 266
Bandwidth 20 (%)
(pkts matched/bytes matched) 0/0
(depth/total drops/no-buffer drops) 0/0/0
exponential weight: 9
mean queue depth: 0

dscp	Transmitted pkts/bytes	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark prob
af11	0/0	0/0	0/0	32	40	1/10
af12	0/0	0/0	0/0	28	40	1/10
af13	0/0	0/0	0/0	24	40	1/10
af21	0/0	0/0	0/0	32	40	1/10

af22	0/0	0/0	0/0	28	40	1/10
af23	0/0	0/0	0/0	24	40	1/10
af31	0/0	0/0	0/0	32	40	1/10
af32	0/0	0/0	0/0	28	40	1/10
af33	0/0	0/0	0/0	24	40	1/10
af41	0/0	0/0	0/0	32	40	1/10
af42	0/0	0/0	0/0	28	40	1/10
af43	0/0	0/0	0/0	24	40	1/10
cs1	0/0	0/0	0/0	22	40	1/10
cs2	0/0	0/0	0/0	24	40	1/10
cs3	0/0	0/0	0/0	26	40	1/10
cs4	0/0	0/0	0/0	28	40	1/10
cs5	0/0	0/0	0/0	30	40	1/10
cs6	0/0	0/0	0/0	32	40	1/10
cs7	0/0	0/0	0/0	34	40	1/10
ef	0/0	0/0	0/0	36	40	1/10
rsvp	0/0	0/0	0/0	36	40	1/10
default	0/0	0/0	0/0	20	40	1/10

Class-map: bronze (match-all)
0 packets, 0 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: ip dscp af31 af32 af33
Queueing
Output Queue: Conversation 267
Bandwidth 20 (%)
(pkts matched/bytes matched) 0/0
(depth/total drops/no-buffer drops) 0/0/0
exponential weight: 9
mean queue depth: 0

dscp	Transmitted pkts/bytes	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark prob
af11	0/0	0/0	0/0	32	40	1/10
af12	0/0	0/0	0/0	28	40	1/10
af13	0/0	0/0	0/0	24	40	1/10
af21	0/0	0/0	0/0	32	40	1/10
af22	0/0	0/0	0/0	28	40	1/10
af23	0/0	0/0	0/0	24	40	1/10
af31	0/0	0/0	0/0	32	40	1/10
af32	0/0	0/0	0/0	28	40	1/10
af33	0/0	0/0	0/0	24	40	1/10
af41	0/0	0/0	0/0	32	40	1/10
af42	0/0	0/0	0/0	28	40	1/10
af43	0/0	0/0	0/0	24	40	1/10

cs1	0/0	0/0	0/0	22	40	1/10
cs2	0/0	0/0	0/0	24	40	1/10
cs3	0/0	0/0	0/0	26	40	1/10
cs4	0/0	0/0	0/0	28	40	1/10
cs5	0/0	0/0	0/0	30	40	1/10
cs6	0/0	0/0	0/0	32	40	1/10
cs7	0/0	0/0	0/0	34	40	1/10
ef	0/0	0/0	0/0	36	40	1/10
rsvp	0/0	0/0	0/0	36	40	1/10
default	0/0	0/0	0/0	20	40	1/10

Class-map: best-effort (match-all)
3346 packets, 214842 bytes
5 minute offered rate 0 bps, drop rate 0 bps
Match: ip dscp default
police:
56000 bps, 1750 limit, 1750 extended limit
conformed 3346 packets, 214842 bytes; action: transmit
exceeded 0 packets, 0 bytes; action: drop
violated 0 packets, 0 bytes; action: drop
conformed 0 bps, exceed 0 bps, violate 0 bps

Class-map: class-default (match-any)
128225 packets, 12224993 bytes
5 minute offered rate 1000 bps, drop rate 0 bps
Match: any

L. PHB POLICY-MAP FOR ALL TRAFFIC ON MAR_REGT

MAR_REGT>sh policy int e1/3
Ethernet1/3

Service-policy output: JMNO

Class-map: platinum (match-all)
247004 packets, 296365418 bytes
5 minute offered rate 13000 bps, drop rate 0 bps
Match: ip dscp ef
Queueing
Strict Priority
Output Queue: Conversation 264
Bandwidth 256 (kbps) Burst 6400 (Bytes)
(pkts matched/bytes matched) 17/15716
(total drops/bytes drops) 0/0

Class-map: gold (match-all)
1645978 packets, 568163287 bytes
5 minute offered rate 55000 bps, drop rate 0 bps
Match: ip dscp af11 af12 af13
Queueing
Output Queue: Conversation 137
Bandwidth 35 (%)
(pkts matched/bytes matched) 78/26186
(depth/total drops/no-buffer drops) 0/0/0
exponential weight: 9
mean queue depth: 0

dscp	Transmitted pkts/bytes	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark prob
af11	895305/338884279	0/0	0/0	32	40	1/10
af12	750675/229279312	0/0	0/0	28	40	1/10
af13	0/0	0/0	0/0	24	40	1/10
af21	0/0	0/0	0/0	32	40	1/10
af22	0/0	0/0	0/0	28	40	1/10
af23	0/0	0/0	0/0	24	40	1/10
af31	0/0	0/0	0/0	32	40	1/10
af32	0/0	0/0	0/0	28	40	1/10
af33	0/0	0/0	0/0	24	40	1/10
af41	0/0	0/0	0/0	32	40	1/10
af42	0/0	0/0	0/0	28	40	1/10
af43	0/0	0/0	0/0	24	40	1/10
cs1	0/0	0/0	0/0	22	40	1/10
cs2	0/0	0/0	0/0	24	40	1/10
cs3	0/0	0/0	0/0	26	40	1/10
cs4	0/0	0/0	0/0	28	40	1/10
cs5	0/0	0/0	0/0	30	40	1/10
cs6	0/0	0/0	0/0	32	40	1/10
cs7	0/0	0/0	0/0	34	40	1/10
ef	0/0	0/0	0/0	36	40	1/10
rsvp	0/0	0/0	0/0	36	40	1/10
default	0/0	0/0	0/0	20	40	1/10

Class-map: silver (match-all)
789514 packets, 373474181 bytes
5 minute offered rate 35000 bps, drop rate 0 bps
Match: ip dscp af21 af22 af23
Queueing
Output Queue: Conversation 138
Bandwidth 20 (%)
(pkts matched/bytes matched) 34/15466

(depth/total drops/no-buffer drops) 0/0/0
exponential weight: 9
mean queue depth: 0

dscp	Transmitted pkts/bytes	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark prob
af11	0/0	0/0	0/0	32	40	1/10
af12	0/0	0/0	0/0	28	40	1/10
af13	0/0	0/0	0/0	24	40	1/10
af21	466123/289865035	0/0	0/0	32	40	1/10
af22	323424/83627328	0/0	0/0	28	40	1/10
af23	0/0	0/0	0/0	24	40	1/10
af31	0/0	0/0	0/0	32	40	1/10
af32	0/0	0/0	0/0	28	40	1/10
af33	0/0	0/0	0/0	24	40	1/10
af41	0/0	0/0	0/0	32	40	1/10
af42	0/0	0/0	0/0	28	40	1/10
af43	0/0	0/0	0/0	24	40	1/10
cs1	0/0	0/0	0/0	22	40	1/10
cs2	0/0	0/0	0/0	24	40	1/10
cs3	0/0	0/0	0/0	26	40	1/10
cs4	0/0	0/0	0/0	28	40	1/10
cs5	0/0	0/0	0/0	30	40	1/10
cs6	0/0	0/0	0/0	32	40	1/10
cs7	0/0	0/0	0/0	34	40	1/10
ef	0/0	0/0	0/0	36	40	1/10
rsvp	0/0	0/0	0/0	36	40	1/10
default	0/0	0/0	0/0	20	40	1/10

Class-map: bronze (match-all)
1348688 packets, 390527705 bytes
5 minute offered rate 33000 bps, drop rate 0 bps
Match: ip dscp af31 af32 af33
Queueing
Output Queue: Conversation 139
Bandwidth 20 (%)
(pkts matched/bytes matched) 60/16728
(depth/total drops/no-buffer drops) 0/0/0
exponential weight: 9
mean queue depth: 0

dscp	Transmitted pkts/bytes	Random drop pkts/bytes	Tail drop pkts/bytes	Minimum thresh	Maximum thresh	Mark prob
af11	0/0	0/0	0/0	32	40	1/10
af12	0/0	0/0	0/0	28	40	1/10

af13	0/0	0/0	0/0	24	40	1/10
af21	0/0	0/0	0/0	32	40	1/10
af22	0/0	0/0	0/0	28	40	1/10
af23	0/0	0/0	0/0	24	40	1/10
af31	698865/204952263	0/0	0/0	32	40	1/10
af32	649876/185592120	0/0	0/0	28	40	1/10
af33	0/0	0/0	0/0	24	40	1/10
af41	0/0	0/0	0/0	32	40	1/10
af42	0/0	0/0	0/0	28	40	1/10
af43	0/0	0/0	0/0	24	40	1/10
cs1	0/0	0/0	0/0	22	40	1/10
cs2	0/0	0/0	0/0	24	40	1/10
cs3	0/0	0/0	0/0	26	40	1/10
cs4	0/0	0/0	0/0	28	40	1/10
cs5	0/0	0/0	0/0	30	40	1/10
cs6	0/0	0/0	0/0	32	40	1/10
cs7	0/0	0/0	0/0	34	40	1/10
ef	0/0	0/0	0/0	36	40	1/10
rsvp	0/0	0/0	0/0	36	40	1/10
default	0/0	0/0	0/0	20	40	1/10

Class-map: best-effort (match-all)
471861 packets, 261687143 bytes
5 minute offered rate 41000 bps, drop rate 0 bps
Match: ip dscp default
police:
56000 bps, 1750 limit, 1750 extended limit
conformed 471640 packets, 261347677 bytes; action: transmit
exceeded 132 packets, 186736 bytes; action: drop
violated 117 packets, 168242 bytes; action: drop
conformed 17000 bps, exceed 0 bps, violate 0 bps

Class-map: class-default (match-any)
12948 packets, 1311105 bytes
5 minute offered rate 1000 bps, drop rate 0 bps
Match: any

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] JMNO Working Group, “Joint Mobile Network Operations: Tactics, Techniques, and Procedures,” Draft Version 1.8. Quantico, Virginia, June 2007.

[2] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-down Approach Featuring the Internet*, Third Edition. Boston: Addison-Wesley, 2005.

[3] Cisco Systems, Inc., “Enhanced Interior Gateway Routing Protocol.” http://www.cisco.com/warp/public/103/eigrp-toc.html, Last visited August 2007.

[4] Y. Rekhter and T. Li, *RFC 1771: A Border Gateway Protocol 4*. Internet Engineering Task Force, Network Working Group, 1995.

[5] Cisco Systems, Inc., “Border Gateway Protocol.” http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/bgp.htm, Last visited August 2007.

[6] R. Droms, *RFC 2131: Dynamic Host Configuration Protocol*. Internet Engineering Task Force, Network Working Group, 1997.

[7] Cisco Systems, Inc., “Cisco IOS Quality of Service Solutions Configuration Guide,” Release 12.2. http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/, 2004, Last visited August 2007.

[8] Wu-Chang Feng, Dilip D. Kandlur, Debanjan Saha, and Kang G. Shin, "Adaptive Packet Marking for Maintaining End-to-End Throughput in a Differentiated-Services Internet," *IEEE/ACM Transactions on Networking,* vol. 7, no. 5, pp. 685-697, October 1999.

[9] S. Shenker, C. Partridge, and R. Guerin, *RFC 2212: Specification of Guaranteed Quality of Service*. Internet Engineering Task Force, Network Working Group, 1997.

[10] J. Wroclawski, *RFC 2211: Specification of the Controlled-Load Network Element Service*. Internet Engineering Task Force, Network Working Group, 1997.

[11] Ahsan Habib, Sonia Fahmy, and Bharat Bhargava, "Monitoring and Controlling QoS Network Domains," *International Journal of Network Management,* vol. 15, pp. 11-29, November 2004.

[12] A. Beben, "EQ-BGP: An Efficient Inter-domain QoS Routing Protocol," Warsaw University of Technology, http://tnt.tele.pw.edu.pl/include/members/Artikuly/beben-eqbgp.pdf, Last visited August 2007.

[13] Shengquan Wang, Dong Xuan, Riccardo Bettati, and Wei Zhao, "Providing Absolute Differentiated Services for Real-Time Applications in Static- Priority Scheduling Networks," *IEEE/ACM Transactions on Networking*, Vol. 12, No. 2, pp. 326-339, April 2004.

[14] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, *RFC 2475: An Architecture for Differentiated Services*. Internet Engineering Task Force, Network Working Group, 1998.

[15] J. Babiarz, K. Chan, and F. Baker, *RFC 4594: Configuration Guidelines for DiffServ Service Classes*. Internet Engineering Task Force, Network Working Group, 2006.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

4. Director, Training and Education
   MCCDC, Code C46
   Quantico, Virginia

5. Director, Marine Corps Research Center
   MCCDC, Code C40RC
   Quantico, Virginia

6. Director, Marine Corps Tactical Systems Support Activity
   (Attn: Operations Officer)
   Camp Pendleton, California

7. Director, Joint Mobile Network Operations
   Quantico, Virginia

8. Professor Geoffrey Xie
   Naval Postgraduate School
   Monterey, California

9. Professor John Gibson
   Naval Postgraduate School
   Monterey, California